

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

GUSTAVO HOMMERDING ALT

**Controle em Tempo Real de Robôs
através de Redes IP**

Trabalho de Conclusão apresentado como
requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Marcelo Soares Lubaszewski
Orientador

Prof. Dr. Walter Fetter Lages
Co-orientador

Porto Alegre, dezembro de 2003

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Alt, Gustavo Hommerding

Controle em Tempo Real de Robôs através de Redes IP / Gustavo Hommerding Alt. – Porto Alegre: PPGC da UFRGS, 2003.

87 f.: il.

Trabalho de Conclusão (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2003. Orientador: Marcelo Soares Lubaszewski; Co-orientador: Walter Fetter Lages.

1. Controle de Robôs. 2. Rede de Controle. 3. Controle em Tempo Real. 4. Internet. 5. Rede de computadores. I. Lubaszewski, Marcelo Soares. II. Lages, Walter Fetter. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof^a. Wrana Maria Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitora Adjunta de Pós-Graduação: Prof^a. Jocélia Grazia

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

DEDICATÓRIA

Este trabalho dedico à Marise, minha sempre amada esposa, por sua grande paciência no decorrer deste longo período de ausência, por compreender, apoiar e incentivar nesta empreitada. Mesmo longe, o amor sempre prevaleceu e como resultado permitiu chegar ao final deste trabalho.

Também dedico à minha família, por seu apoio durante todo o período deste desafio.

A meu co-orientador, Prof. Walter, agradeço por compartilhar seus conhecimentos ao longo deste período de convivência acadêmica.

AGRADECIMENTOS

O autor agradece à Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS), pelo apoio a este projeto através do Programa de Apoio ao Desenvolvimento Científico e Tecnológico em Informática (PROADI), processo nº 01/05670.0.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	9
LISTA DE FIGURAS	11
LISTA DE TABELAS	13
RESUMO	15
ABSTRACT	17
1 INTRODUÇÃO	19
1.1 Motivação	20
1.2 Objetivo	21
1.3 Organização do texto	21
2 DESCRIÇÃO DO SISTEMA	23
2.1 Hardware utilizado	23
2.2 Descrição do sistema RTAI	25
2.3 Descrição das métricas	26
2.3.1 Características da rede	26
2.3.2 Características do sistema de controle	30
2.3.3 Características calculadas da temporização de rede	31
2.3.4 Características de desempenho do sistema de controle calculadas	32
3 CONTROLE LOCAL	33
3.1 Modelo do robô	33
3.2 Lei de controle	35
3.3 Implementação	36
3.4 Resultados	37
4 CONTROLE ATRAVÉS DE REDE IP	43
4.1 Descrição da rede	43
4.1.1 Rede Ethernet	45
4.1.2 Protocolos UDP	45
4.1.3 Leitura do datagrama	46
4.2 Manipulador robótico - daemon robô	46
4.3 Controlador	49

4.4	Protocolo de Aplicação	50
4.4.1	Pacotes do controlador	50
4.4.2	Pacote transmitido pelo manipulador robótico	52
4.5	Inserção de atrasos	53
4.6	Resultados	54
4.6.1	Controle através da rede sem atraso	54
4.6.2	Controle através da rede com atraso	59
5	CONTROLE COM COMPENSAÇÃO DOS ATRASOS ROBÔ / CONTROLADOR	65
5.1	Medida dos atrasos	66
5.2	Atraso do sinal do sensor	66
5.3	Resultados	67
6	CONTROLE COM COMPENSAÇÃO DOS ATRASOS ROBÔ / CONTROLADOR E CONTROLADOR / ROBÔ	73
6.1	Atraso do sinal de controle	73
6.2	Resultados	74
7	CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS	81
	REFERÊNCIAS	85

LISTA DE ABREVIATURAS E SIGLAS

ATM	Asynchronous Transfer Mode
BACnet	Building Automation and Control Network
CAN	Controller Area Network
CPU	Central Processor Unit
CRC	Cyclic Redundance Checking
CSMA	Carrier Sense Multiple Access
CSMA/BA	Carrier Sense Multiple Access with Bitwise Arbitration
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DNS	Domain Name System
FIFO	First-In First-Out
GPS	Global Position System
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISA	Industry Standard Architecture
LAN	Local Area Network
MAP	Manufacturing Automation Protocol
NIC	Network Interface Card
NFS	Network File System
NTP	Network Time Protocol
PC	Personal Computer
PID	Proporcional+Integral+Derivativo
PWM	Pulse Width Modulation
QoS	Quality of Service

RAM	Random Access Memory
RTAI	Real-Time Application Interface
SNMP	Simple Network Management Protocol
UDP	User Datagram Protocol
WWW	World Wide Web

LISTA DE FIGURAS

Figura 2.1:	Robô Janus.	24
Figura 2.2:	Esquemático do manipulador robótico.	24
Figura 2.3:	Parte do programa RTAI utilizado para implementar a temporização das tarefas.	27
Figura 2.4:	Parte do programa RTAI utilizado para obter o tempo de execução da função <i>get_time()</i>	28
Figura 2.5:	Tempo de execução da função <i>get_time()</i>	29
Figura 2.6:	Instantes de tempo coletados nos experimentos.	29
Figura 2.7:	Resposta do sistema de controle.	30
Figura 3.1:	Estrutura do controle por torque calculado.	36
Figura 3.2:	Período de recepção do sinal do sensor com controle local.	37
Figura 3.3:	Período de envio do sinal de controle com controle local.	38
Figura 3.4:	Tempo de <i>round-trip</i> com controle local.	39
Figura 3.5:	Resposta do controlador (posição angular em graus, tempo em segundos).	40
Figura 4.1:	Estrutura do sistema de controle baseado em rede.	43
Figura 4.2:	Protocolos de rede utilizados para o controle do robô através da rede.	44
Figura 4.3:	Comunicação controlador / robô através do protocolo UDP.	47
Figura 4.4:	Esquema utilizando manipulador robótico temporizado.	48
Figura 4.5:	Formato do pacote genérico de controle.	50
Figura 4.6:	Formato do pacote de controle.	50
Figura 4.7:	Formato do pacote Liga.	51
Figura 4.8:	Formato do pacote Desliga.	52
Figura 4.9:	Formato do pacote do manipulador robótico.	52
Figura 4.10:	Filtro para inserção de atrasos.	53
Figura 4.11:	Período de recepção do sinal de sensor com controle através da rede sem atraso.	55
Figura 4.12:	Período de recepção do sinal de controle com controle através da rede sem atraso.	56
Figura 4.13:	Tempo de <i>round-trip</i> com controlador com controle através da rede sem atraso.	57
Figura 4.14:	Resposta do controlador (posição angular em graus, tempo em segundos).	58

Figura 4.15: Período de recepção do sinal de sensor com controle através da rede com atraso.	60
Figura 4.16: Período de recepção do sinal de controle com controle através da rede com atraso.	61
Figura 4.17: Tempo de <i>round-trip</i> com controle através da rede com atraso. . .	62
Figura 4.18: Resposta do controlador.	63
Figura 5.1: Estrutura do sistema de controle com compensação.	65
Figura 5.2: Período de recepção do sinal de controle com compensação de atraso do sinal do sensor.	67
Figura 5.3: Período de recepção do sinal de controle com compensação de atraso do sinal do sensor.	68
Figura 5.4: Tempo de <i>round-trip</i> com compensação do atraso do sinal do sensor. 69	
Figura 5.5: Resposta do controlador com compensação dos atrasos robô / controlador.	70
Figura 6.1: Controle com compensação dos atrasos robô/controlador e do atraso controlador/robô.	73
Figura 6.2: Período de recepção do sinal de sensor com compensações dos atrasos robô / controlador e controlador / robô.	75
Figura 6.3: Período de recepção do sinal de controle com compensações dos atrasos robô/controlador e controlador/robô.	76
Figura 6.4: Tempo de <i>round-trip</i> com compensações dos atrasos robô / controlador e controlador / robô.	77
Figura 6.5: Resposta do controlador com compensações no controlador e no robô.	78
Figura 7.1: Comparação das estratégias de controle pelo critério ITAE.	83

LISTA DE TABELAS

Tabela 2.1:	Estatísticas do tempo de execução da função <i>get_time()</i>	26
Tabela 3.1:	Estatísticas das métricas com controle local.	40
Tabela 3.2:	Desempenho do sistema de controle com controle local.	41
Tabela 4.1:	Descrição do pacote genérico de controle.	50
Tabela 4.2:	Descrição do pacote de controle.	51
Tabela 4.3:	Descrição do pacote Liga.	51
Tabela 4.4:	Descrição do pacote Desliga.	52
Tabela 4.5:	Descrição do pacote de leitura dos sensores.	52
Tabela 4.6:	Estatísticas das métricas com controle através da rede sem atraso.	58
Tabela 4.7:	Desempenho do sistema de controle através da rede sem atraso.	59
Tabela 4.8:	Estatísticas das métricas com controle através da rede com atraso.	59
Tabela 4.9:	Desempenho do sistema de controle através da rede com atraso.	64
Tabela 5.1:	Estatísticas das métricas com controle através da rede com compensação de atraso do sinal do sensor.	69
Tabela 5.2:	Desempenho do sistema de controle com compensação dos atrasos robô/controlador.	71
Tabela 6.1:	Estatísticas das métricas computadas para o controle com rede carregada com compensações no controlador e no robô.	77
Tabela 6.2:	Desempenho do sistema de controle com compensações dos atrasos robô/controlador e controlador/robô.	79
Tabela 7.1:	Desempenho dos sistemas de controle através da rede.	82

RESUMO

Com o intuito de utilizar uma rede com protocolo IP para a implementação de malhas fechadas de controle, este trabalho propõe-se a realizar um estudo da operação de um sistema de controle dinâmico distribuído, comparando-o com a operação de um sistema de controle local convencional. Em geral, a decisão de projetar uma arquitetura de controle distribuído é feita baseada na simplicidade, na redução dos custos e confiabilidade; portanto, um diferencial bastante importante é a utilização da rede IP. O objetivo de uma rede de controle não é transmitir dados digitais, mas dados analógicos amostrados. Assim, métricas usuais em redes de computadores, como quantidade de dados e taxa de transferências, tornam-se secundárias em uma rede de controle. São propostas técnicas para tratar os pacotes que sofrem atrasos e recuperar o desempenho do sistema de controle através da rede IP. A chave para este método é realizar a estimação do conteúdo dos pacotes que sofrem atrasos com base no modelo dinâmico do sistema, mantendo o sistema com um nível adequado de desempenho. O sistema considerado é o controle de um manipulador antropomórfico com dois braços e uma cabeça de visão estéreo totalizando 18 juntas. Os resultados obtidos mostram que se pode recuperar boa parte do desempenho do sistema.

Palavras-chave: Controle de Robôs, Rede de Controle, Controle em Tempo Real, Internet, Rede de computadores.

Real-Time Robot Control over IP Networks

ABSTRACT

In order to enable the use of IP networks to implement closed-loop control systems, this work compares a distributed control system to its conventional counterpart, which is based on local control. Usually the option for a distributed control architecture is motivated by simplicity, cost reduction and reliability, hence the use of an IP network. The goal of a control network is not to transmit pure digital data, but to transmit digitized control and sensor signals. Therefore, metrics common in computer networks, such as amount of transferred data and transfer rates, become of little importance. This work proposes a method to compensate for the delays imposed to network packets and to recover the control system performance. The key for this method is to estimate the contents of delayed network packets based on the system dynamic model. The example system is an antropomorphic robot with two arms and a stereo vision head. Results show that the system performance can be recovered.

Keywords: Robot control, Control Network, Real-Time Control, Computer Network.

1 INTRODUÇÃO

Com a popularização da rede Internet, há um interesse muito grande por implementar novas aplicações neste tipo de rede. Fazendo-se uso das novas funcionalidades e facilidades, não se restringe mais a utilização da aplicação apenas a um local físico, permitindo sua utilização no âmbito global envolvendo uma variedade de sistemas e os mais diversos meios de transmissão existentes. Atualmente, a teleoperação tem sido intensamente estudada, motivada pela ampla gama de aplicações existentes que emergem em diversos campos de atuação, como por exemplo a utilização em plantas nucleares (HIRZINGER et al., 1993), no espaço para operação de veículos em lugares inóspitos (HARA; HIRUKAWA, 2001), operação de telescópios (BEJCZY, 1996), aplicações médicas (FUNDA et al., 1996), entre outras.

Em geral, redes para sistemas de controle diferem de redes de computadores em alguns aspectos importantes e que necessitam serem levados em consideração. Uma rede de dados possui grandes pacotes de dados, transmissões em rajadas que não possuem uma frequência definida e elevadas taxas de dados na rede. Redes de controle, em contrapartida, tem por característica pacotes pequenos com exigências de tempo real (ZHANG; BRANICKY; PHILLIPS, 2001). O elemento chave que distingue a rede de computadores em geral de uma rede dedicada a sistemas de controle é a capacidade de suportar aplicações de tempo real ou de missão-crítica. Um aspecto importante a considerar é que o objetivo em uma rede de controle não é transmitir dados digitais, mas dados analógicos amostrados. Assim, métricas usuais em redes de computadores, como quantidade de dados e taxa de transferências, tornam-se secundárias em uma rede de controle. Outro fator a considerar é que em uma rede de controle, ao contrário do que pode parecer em um primeiro momento, a principal fonte dos atrasos de transporte não é o meio físico. A taxa de transferência da rede, em geral, é tal que o tempo de transmissão dos dados é muito pequeno. O atraso de transporte é gerado basicamente pelas filas existentes no sistema de rede. Em aplicações usuais de rede de computadores, estas filas otimizam o tempo médio de atraso dos pacotes; porém em uma rede de sistema de controle, muitas vezes, é mais conveniente eliminar estas filas (WALSH; YE, 2001). Para o protocolo Internet, o tempo de atraso não pode ser previsto, é aleatório e segue com um modelo estocástico muito complexo. Várias tentativas têm sido feitas para modelar o atraso de rede que ocorre através da Internet e vários modelos complexos tem sido obtidos (RIEDI et al., 1999).

Existem várias diferenças entre o sistema de controle remoto através da Internet e o sistema de controle através de meio privado. As tecnologias padrões utilizadas

em redes de computadores, no nível de enlace e físico, tem sido adaptadas para este contexto, como *daisy-chained* RS-232, *multi-drop* RS-485 (BROCKETT, 1995), IEEE-488 e Ethernet e suas extensões, como *wireless* Ethernet IEEE-802.11 (YE; WALSH; BUSHNELL, 2000). Adicionalmente, redes especializadas para controle e automação tem sido desenvolvidas, como CAN para automação industrial, BACNet para automação predial, Profibus (TOVAR; VASQUES, 1999) e Foundation Fieldbus para controle de processos (BUSHNELL, 2001).

Redes de controle devem obedecer a dois critérios principais: tempo de atraso limitado e transmissão garantida. A utilização de diversos protocolos tem sido proposta para redes de controle, satisfazendo estas exigências. Entre eles pode-se citar Ethernet (CSMA/CD (IEEE-802.3)), token bus (IEEE-802.4), token ring (IEEE-802.5) e CAN (CSMA/BA). O importante aqui é o método utilizado por cada um destes protocolos para determinar qual dispositivo em uma rede de controle tem o direito de acessar o meio de transmissão. Obviamente, este método tem grande influência sobre o atraso imposto às mensagens. Embora alguns protocolos sejam determinísticos em relação ao meio de transmissão, como os protocolos token bus (IEEE-802.4), token ring (IEEE-802.5) e CAN (CSMA/BA), neste trabalho optou-se por utilizar, para o nível de enlace e físico, a rede Ethernet, que utiliza um protocolo não determinístico. Esta escolha deve-se, assim como a opção pelo protocolo IP, ao baixo custo e a alta disponibilidade no mercado de dispositivos e equipamentos deste padrão (KWEON; SHIN, 2003). Importante notar que o protocolo Ethernet apresenta um desempenho superior ao apresentado pelos demais protocolos citados acima, quando a rede está pouco carregada (LIAN; MOYNE; TILBURY, 2001). Soluções tem sido propostas neste sentido (SKEIE; JOHANNESSEN; BRUNNER, 2002), como a utilização da Ethernet no meio industrial (CASEY, 1990), bem como comparações e avaliações da utilização da rede Ethernet (RAMAKRISHNAN; YANG, 1994). No entanto, considerando que as suas principais desvantagens estão relacionadas com o controle de acesso ao meio através de CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*), o uso de equipamentos *switches* (equipamentos que realizam o chaveamento de portas lógicas) reduzem bastante o problema (TANENBAUM, 1997). Considerando a redução de custo observada neste tipo de equipamento, esta torna-se uma solução interessante.

1.1 Motivação

O projeto de sistemas de controle distribuído utilizando redes convencionais é novo para a comunidade acadêmica de controle, embora não seja tão novo na indústria (BUSHNELL, 2001), onde são utilizados em plantas de produção automatizada, bem como na aviação e em automóveis. Esta peculiaridade permite supor que ainda exista bastante espaço para desenvolvimentos nesta área.

A principal motivação neste trabalho está em utilizar redes de computadores convencionais, baseadas no protocolo IP, para operar sistemas de controle distribuídos. Com isso, pretende-se operar o sistema através do uso de uma tecnologia mais difundida, aumentar a disponibilidade através de uma rede compartilhada e popularizar a utilização de sistemas de controle, não restringindo a operação em ambientes isolados e fechados. Em geral, a decisão de projetar uma arquitetura de controle distribuído

é feita baseado na simplicidade, redução dos custos de instalação, manutenção e confiabilidade; portanto, um diferencial bastante importante é a utilização da rede IP.

1.2 Objetivo

O trabalho tem por objetivo, avaliar técnicas que permitam garantir níveis de qualidade adequados à implementação de sistemas distribuídos de controle industrial, utilizando redes IP. Igualmente, pretende-se estudar os efeitos da variação dos atrasos de tempo em sistemas de controle baseados neste tipo de rede e o desenvolvimento de técnicas que possam assegurar estabilidade e desempenho do sistema, apesar dos atrasos sofridos pelos pacotes de rede. Como plataforma de testes, será utilizado o manipulador robótico Janus, disponível no Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul.

1.3 Organização do texto

O restante do texto está organizado da seguinte maneira:

No capítulo 2, apresenta-se como foram implementadas as aplicações de controle e do manipulador robótico, a descrição da rede, do sistema de *hardware* e *software* e as métricas utilizadas nos experimentos.

No capítulo 3, é apresentado o controle local sem a rede incorporada. Este estudo tem por objetivo analisar o caso ideal de operação do controlador e do manipulador robótico sem a presença da rede, baseado em um sistema de controle convencional. Este é um parâmetro de referência a ser alcançado quando a rede estiver incorporada entre o controlador remoto e o manipulador robótico.

No capítulo 4, é apresentado o controle em rede sem qualquer técnica de compensação de atrasos. Este estudo tem por objetivo analisar o desempenho do controlador através da rede, bem como com atrasos nos pacotes que trafegam no sentido do manipulador robótico para o controlador e no sentido contrário.

No capítulo 5, é apresentado o controle com compensação apenas do atraso manipulador robótico / controlador. Este estudo tem por objetivo recuperar o desempenho do controlador local sem rede, impondo uma compensação dos atrasos dos pacotes que trafegam na rede.

No capítulo 6, é apresentado o controle com compensações de atraso do manipulador robótico / controlador e controlador / manipulador robótico. Este estudo tem por objetivo recuperar o desempenho do sistema de controle, segundo o desempenho obtido com o controlador local. Com os resultados obtidos, busca-se caracterizar o comportamento deste sistema em relação ao controle local sem rede.

No capítulo 7, são apresentados as conclusões sobre os métodos propostos e sugestões para trabalhos futuros.

2 DESCRIÇÃO DO SISTEMA

Neste capítulo, descreve-se o sistema robô. Apresenta-se como foram implementadas as aplicações de controle e do manipulador, com descrição da rede, bem como do sistema de *hardware*, *software* e das métricas utilizadas.

O sistema de controle distribuído é formado pelo manipulador robótico e o controlador, que está situado em algum ponto da rede, com a malha fechada através da rede IP. O manipulador robótico utilizado neste trabalho é o robô Janus, disponível no Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul. Este manipulador é dotado de dois braços com 8 graus de liberdade cada e uma cabeça de visão estéreo com 3 graus de liberdade, como mostrado na figura 2.1.

2.1 Hardware utilizado

A configuração do hardware deste sistema de controle inclui um robô, uma estação servidora e uma estação de controle em algum ponto da rede.

Os programas do controlador e do manipulador robótico (indicados na figura 2.1), são executados em computadores PC com sistema operacional Linux RTAI. A motivação por utilizar este sistema está nas suas características de disponibilidade, facilidade para se implementar *drivers*, sendo um sistema extremamente robusto e flexível.

A estação que executa o acionamento do manipulador robótico é composta de um computador PC com processador AMD K6 400 MHz e 152 MB de memória RAM.

A estação que executa o controle é composto de um computador PC com processador Celeron 400 MHz e 128 MB de memória RAM.

A estação que executa a função do roteador, com a filtragem dos pacotes e controle dos pacotes é composto de um computador PC com processador Athlon 2,7 GHz e 512 MB de memória RAM.

Todas as estações utilizam o sistema operacional Linux, com *kernel 2.4.20* e *patch RTAI*, versão 24.1.11.

O sistema do manipulador robótico executa as operações de manuseio dos dados do sistema robô, iniciando a operação com entrada do dado do sinal de controle de uma das juntas do robô, fornecendo como saída o dado da posição da respectiva junta (sinal do sensor). O sistema é constituído das seguintes partes:

Estação Servidora: realiza a interface com o controlador e o robô.

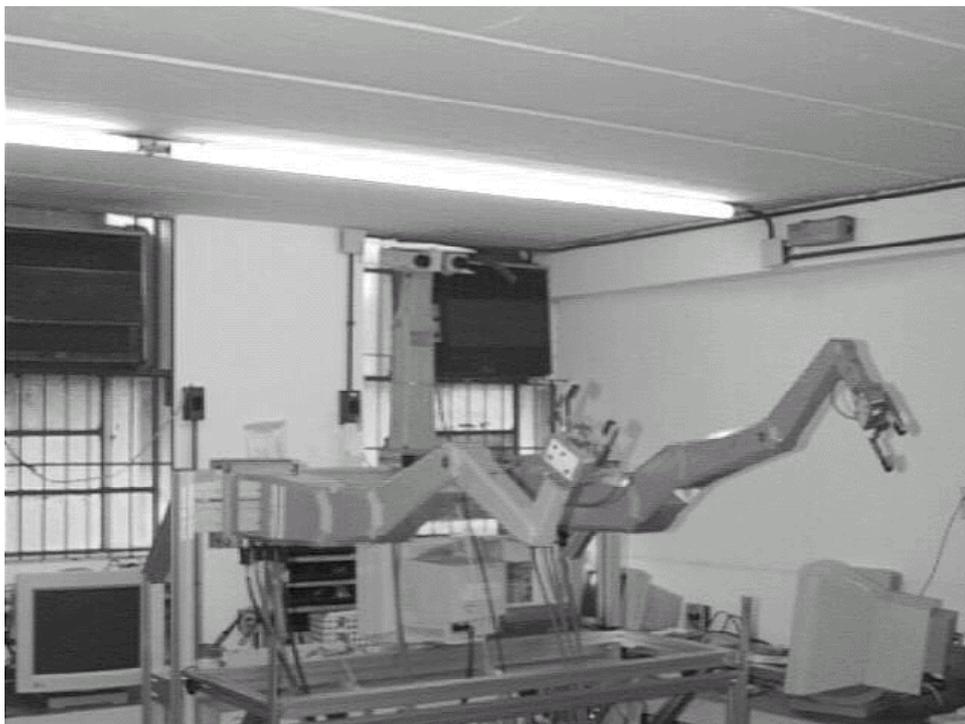


Figura 2.1: Robô Janus.

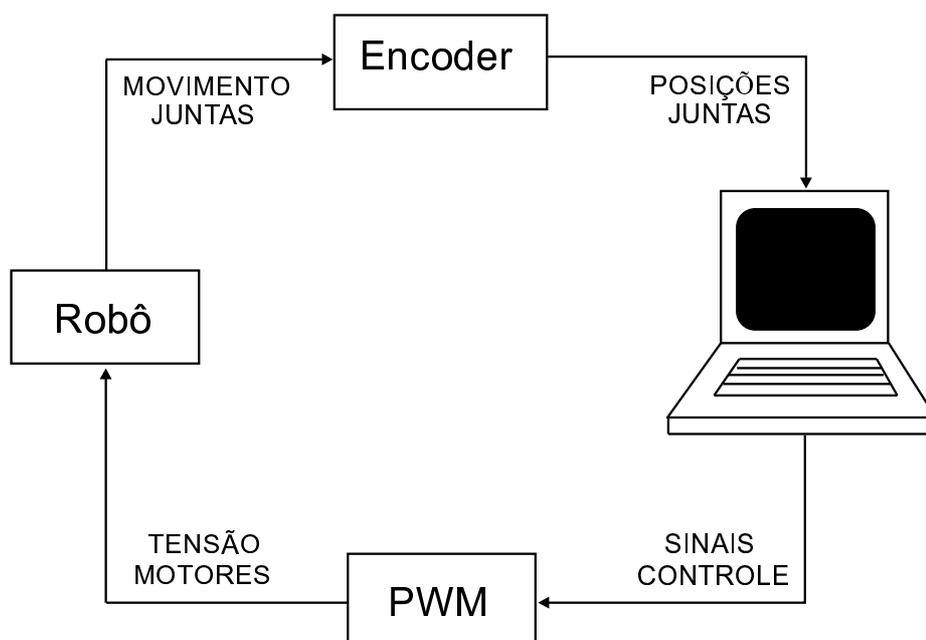


Figura 2.2: Esquemático do manipulador robótico.

PWM (Pulse Width Modulation): recebe os sinais de controle e processa estes convertendo os sinais digitais, recebidos do controle, para sinais analógicos (tensão), que são aplicados nas juntas do robô.

Robô: é constituído por dois braços com 8 graus de liberdade cada e uma cabeça de visão estéreo com 2 graus de liberdade, totalizando 18 juntas (como apresentado na figura 2.1). Cada junta possui um atuador, um *encoder* incremental, um freio eletromagnético e um sensor indutivo de contato.

Encoder: Converte os dados coletados junto aos leitores dos sensores das juntas, de pulsos elétricos para sinais digitais.

2.2 Descrição do sistema RTAI

Como já comentado e detalhado na figura 4.1, o sistema de controle em rede utilizado neste trabalho é composto por dois nodos: o manipulador robótico e o controlador. As funcionalidades de ambos os nodos são implementadas utilizando-se uma variante para tempo real do sistema operacional Linux, denominada RTAI (CLOUTIER et al., 2000; BIANCHI; DOZIO, 2000).

O RTAI suporta quatro tipos de tarefas:

Processos Linux: são os processos comuns do Linux. Executam apenas quando não há tarefa de tempo real para ser executada.

Tarefas de tempo real no *kernel*: são tarefas de tempo real que executam no espaço do *kernel*, com a menor latência possível. Este tipo de tarefa não pode acessar a maioria dos serviços do *kernel* do Linux ou de bibliotecas, como a biblioteca padrão da linguagem C.

Tarefas LXRT *soft real-time*: são tarefas que executam no modo do usuário sem sofrer preempção dos processos do Linux, mas podem sofrer interrupções do *kernel* do Linux e das demais tarefas de tempo real. A vantagem deste tipo de tarefa é que tem-se acesso aos serviços do *kernel* do Linux e de bibliotecas.

Tarefas LXRT *hard real-time*: são tarefas que executam no modo do usuário sem sofrer preempção de processos ou de interrupções do *kernel* do Linux, podendo sofrer preempção apenas de tarefas e interrupções de tempo real com prioridade mais alta. Acessos a serviços do *kernel* do Linux são permitidos através de um mecanismo de chaveamento implícito para o modo *soft real-time*, com retorno ao modo *hard real-time* após o retorno da chamada ao *kernel*.

O desenvolvimento de todo o sistema utilizado neste trabalho, se baseou no sistema Linux por seus inúmeros benefícios, entre eles por ser um *software* livre, de arquitetura aberta, que permite ser instalado em diversas plataformas.

As exigências do tempo das tarefas do sistema, que devem operar na taxa fixa de 100 Hz, são garantidos utilizando o sistema RTAI no modo *hard real-time*, conforme apresentado em parte do código desenvolvido para o sistema de controle (figura 2.3). O sistema RTAI é também utilizado para monitorar o comportamento temporal dos

pacotes de rede. O comportamento temporal e o desempenho de um controlador executado sobre uma rede IP são caracterizados e comparados com aqueles obtidos com um controlador local.

Para as análises e medidas dos experimentos é importante ressaltar que os tempos medidos nestes experimentos foram obtidos através de funções do sistema RTAI. Segue uma análise do tempo dispendido para se obter estas medidas. Para calcular o tempo dispendido pelo sistema operacional foi desenvolvido um programa específico, conforme exemplificado na figura 2.4. Este programa calcula o tempo gasto pelo sistema operacional para realizar medições de tempo. A partir destes dados coletados, são apresentadas as estatísticas na tabela 2.5.

O cálculo para se obter o tempo médio do tempo dispendido pelo sistema operacional para executar a função $get_time()$, foi obtido somando-se a diferença entre a tomada de dois instantes consecutivos de tempo pela função $get_time()$, dividido pelo número de amostras realizadas (no caso, 1000). O *jitter* foi calculado através da variação do intervalo de tempo entre duas diferenças consecutivas da tomada dos tempo da função $get_time()$.

Tabela 2.1: Estatísticas do tempo de execução da função $get_time()$.

Métrica	Média (segundos)	Desvio Padrão (segundos)
$T_{gettime}$	$1576,01 \times 10^{-09}$	$1212,53 \times 10^{-09}$
$J_{gettime}$	$-0,03 \times 10^{-09}$	$1508,59 \times 10^{-09}$

A figura 2.5 apresenta os gráficos do tempo dispendido pelo sistema operacional para a medida do tempo nos experimentos realizados. Verifica-se que o tempo máximo obtido foi de 11 μs . O tempo dispendido pela função $get_time()$ para o sistema com controle local é significativo, tendo em vista que este tempo é da ordem de 10% do tempo total a ser medido. Para um sistema de controle através da rede, a parcela deste tempo é pequena, já que a ordem é de aproximadamente 1%, no pior caso, do tempo total a ser medido. Portanto, este percentual de tempo dispendido não é significativo para as medidas realizadas no sistema de controle em rede.

2.3 Descrição das métricas

2.3.1 Características da rede

Para melhor visualização, na figura 2.6, é mostrado o esquema com o fluxo dos dados entre controlador e robô e os respectivos dados colhidos nos experimentos. As seguintes temporizações foram coletadas utilizando a função $rt_get_time_ns()$ do sistema RTAI.

$t_{ts}(k)$ Instante de transmissão do sinal do sensor no k -ésimo ciclo.

$t_{rs}(k)$ Instante de recepção do sinal do sensor no k -ésimo ciclo.

$t_{tc}(k)$ Instante de transmissão do sinal de controle no k -ésimo ciclo.

$t_{rc}(k)$ Instante de recepção do sinal de controle no k -ésimo ciclo.

```

{
    ...
    unsigned long maintsk_name = nam2num("MAIN");
    struct sched_param mainsched;

    rt_allow_nonroot_hrt();

    mainsched.sched_priority = sched_get_priority_max(SCHED_FIFO) - 1;
    if(sched_setscheduler(0,SCHED_FIFO, &mainsched) == -1 )
    {
        cout << "Error setting scheduler: " << strerror(errno) << "\n";
        return -1;
    }

    RT_TASK *maintsk;
    if(!(maintsk = rt_task_init(maintsk_name, 1, 0, 0)))
    {
        cout << "Can't init master task\n";
        return -1;
    }

    rt_set_one-shot_mode();
    int period = (int) nano2count((RTIME)(ST*1e9));
    start_rt_timer(period);

    rt_make_hard_real_time();

    rt_task_make_periodic(maintsk,rt_get_time()+period,period);

    ...

    for(double t=0; t <=TF && run; t+=ST)
    {
        rt_task_wait_period();
        ...
    }
    ...
}

```

Figura 2.3: Parte do programa RTAI utilizado para implementar a temporização das tarefas.

```

{
    ...

    unsigned long maintsk_name = nam2num("MAIN");
    struct sched_param mainsched;

    rt_allow_nonroot_hrt();

    mainsched.sched_priority = sched_get_priority_max(SCHED_FIFO) - 1;
    if(sched_setscheduler(0,SCHED_FIFO, &mainsched ) == -1 )
    {
        cout << "Error setting scheduler: " << strerror(errno) << "\n";
        return -1;
    }

    RT_TASK *maintsk;
    if(!(maintsk = rt_task_init(maintsk_name, 1, 0, 0)))
    {
        cout << "Can't init master task\n";
        return -1;
    }

    rt_set_oneshot_mode();
    int period = (int) nano2count((RTIME)(ST*1e9));
    start_rt_timer(period);

    rt_make_hard_real_time();

    rt_task_make_periodic(maintsk,rt_get_time()+period,period);

    ...

    for(double t=0; t <=TF && run; t+=ST)
    {
        rt_task_wait_period();

        time_tx=rt_get_time_ns();

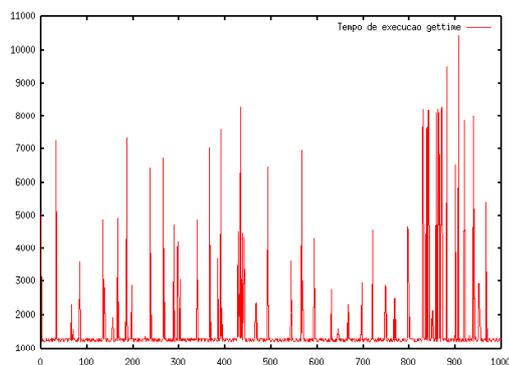
        data[k].time_envio=rt_get_time_ns()-time_tx;

        data[k].temp= t/ST;

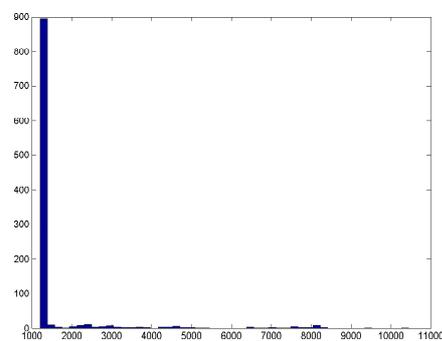
        k++;
    }
    ...
}

```

Figura 2.4: Parte do programa RTAI utilizado para obter o tempo de execução da função *get_time()*.



(a) Tempo de execução (ns).



(b) Histograma do tempo de execução.

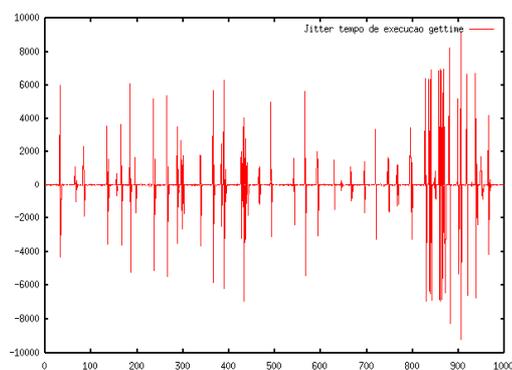
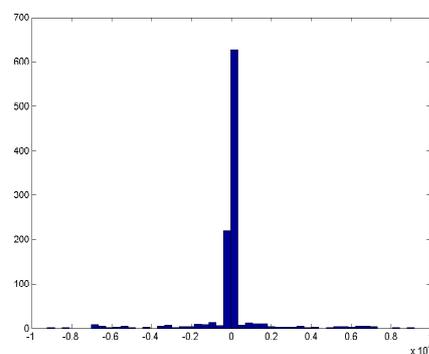
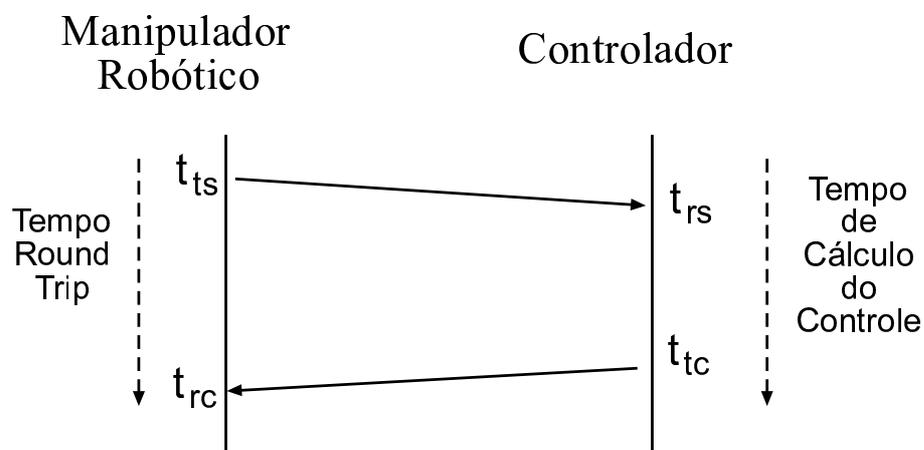
(c) *Jitter* do tempo de execução (ns).(d) Histograma do *jitter*.Figura 2.5: Tempo de execução da função `get_time()`.

Figura 2.6: Instantes de tempo coletados nos experimentos.

2.3.2 Características do sistema de controle

As características básicas da resposta de um sistema de controle são mostrados na figura 2.7

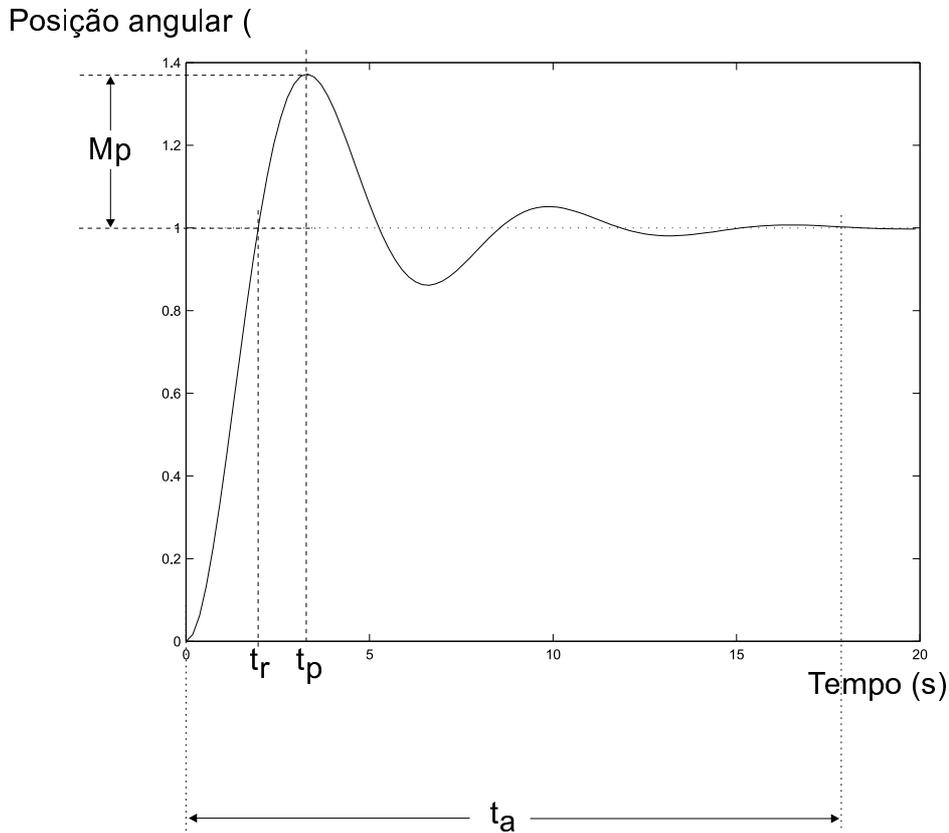


Figura 2.7: Resposta do sistema de controle.

i Identificação da junta.

$\theta_i(k)$ Posição angular da junta.

θ_{r_i} Posição angular de referência da junta.

θ_{p_i} Posição angular máxima da junta.

t_{a_i} Tempo de acomodação. Tempo para o erro tornar-se menor que 3 % do valor de θ_{r_i} .

t_{r_i} Tempo de subida. Tempo para a resposta atingir pela primeira vez o valor de θ_{r_i} .

t_{p_i} Tempo de pico. Tempo para a resposta atingir pela primeira vez o valor de θ_{p_i} .

2.3.3 Características calculadas da temporização de rede

Com base nas definições dos instantes de tempo para transmissão e recepção dos pacotes de rede, conforme apresentado na subseção 2.3.1, é feita a caracterização dos atrasos sofridos pelos pacotes sob as diversas condições de tráfego na rede, como segue:

Período de Recepção do Sinal do Sensor: intervalo de tempo entre duas recepções consecutivas do sinal do sensor, conforme expressão (2.1).

$$\tau_{rs}(k) = t_{rs}(k) - t_{rs}(k - 1) \quad (2.1)$$

Jitter do Período de Recepção do Sinal do Sensor: variação do intervalo de tempo entre duas recepções consecutivas do sinal do sensor, conforme expressão (2.2).

$$J_{rs}(k) = \tau_{rs}(k) - T_s \quad (2.2)$$

onde:

T_s = período da taxa de operação do sistema, ou seja, 10 ms;

Período de Recepção do Sinal de Controle: intervalo de tempo entre duas recepções consecutivas do sinal de controle, conforme expressão (2.3).

$$\tau_{rc}(k) = t_{rc}(k) - t_{rc}(k - 1) \quad (2.3)$$

Jitter do Período de Recepção do Sinal de Controle: variação do intervalo de tempo entre duas recepções consecutivas do sinal de controle, conforme expressão (2.4).

$$J_{rc}(k) = \tau_{rc}(k) - T_s \quad (2.4)$$

Tempo de *round-trip*: período entre a transmissão do sinal de controle correspondente, conforme a expressão (2.5).

$$T_{rt}(k) = t_{rc}(k) - t_{ts}(k) \quad (2.5)$$

Jitter do Tempo de *round-trip*: variação do período entre a transmissão do sinal de controle correspondente, conforme a expressão (2.6).

$$J_{rt}(k) = T_{rt}(k) - \bar{T}_{rt} \quad (2.6)$$

onde:

\bar{T}_{rt} = média do tempo de *round-trip*.

Para cada uma das métricas calculadas, também são computados a média, o desvio padrão e o histograma.

2.3.4 Características de desempenho do sistema de controle calculadas

Para caracterizar o desempenho do sistema de controle, utiliza-se o comportamento da resposta transitória do sistema de controle e índices que refletem o desempenho do sistema. Apresenta-se as métricas utilizadas para avaliar o desempenho do sistema de controle (OGATA, 1985).

Overshoot: Valor de pico da curva de resposta medida a partir do valor de referência da junta (vide figura 2.7).

$$M_p = \frac{(\theta_{p_i} - \theta_{r_i})}{(\theta_{r_i} - \theta_i(0))} \quad (2.7)$$

Critério ISE: o desempenho do sistema é avaliado pela expressão (2.8). A característica deste índice de desempenho é que ele dá grande peso para erros grandes e pequeno peso para erros pequenos. Um sistema que minimiza este critério tende a apresentar uma rápida diminuição em um erro inicial grande. Portanto, a resposta é rápida e oscilatória. Assim, o sistema tem baixa estabilidade relativa.

$$ISE = \int (\theta_{r_i} - \theta_i(t))^2 dt \quad (2.8)$$

Critério IAE: o desempenho do sistema é avaliado pela expressão (2.9). Um sistema ótimo baseado neste critério é um sistema que tem um amortecimento razoável e uma característica de resposta transitória satisfatória.

$$IAE = \int |\theta_{r_i} - \theta_i(t)| dt \quad (2.9)$$

Critério ITSE: o desempenho do sistema é avaliada pela expressão (2.10). Uma característica é que, na resposta ao degrau unitário, um erro inicial grande é ponderado com peso baixo, enquanto que erros que ocorrem mais tarde na resposta transitória são bastante penalizados.

$$ITSE = \int t (\theta_{r_i} - \theta_i(t))^2 dt \quad (2.10)$$

Critério ITAE: o desempenho do sistema é avaliado pela expressão (2.11). Assim como no critério apresentado anteriormente, um erro inicial grande em uma resposta a degrau unitário é ponderado com peso pequeno e erros que ocorrem mais tarde na resposta transitória são bastante penalizados. A característica de um sistema que minimiza este critério é que o *overshoot* é pequeno e oscilações são bem amortecidas.

$$ITAE = \int t |(\theta_{r_i} - \theta_i(t))| dt \quad (2.11)$$

3 CONTROLE LOCAL

Neste capítulo, será apresentado o controle local, sem a rede incorporada entre o controlador e o manipulador robótico. Foram realizados experimentos com objetivo de analisar o desempenho ideal a ser obtido com os métodos para comparar os efeitos dos atrasos introduzidos pela rede. Nesta situação, o controlador e o manipulador robótico são implementados em um único programa, operando como um controlador convencional.

3.1 Modelo do robô

O modelo dinâmico de um manipulador rígido com n graus de liberdade baseado na formulação de Lagrange-Euler (PAUL, 1981; FU; GONZALEZ; LEE, 1987) pode ser escrito na forma:

$$\tau = D(q) \ddot{q} + V(q, \dot{q}) + G(q) \quad (3.1)$$

onde:

q = vetor de coordenadas generalizadas;

τ = vetor de forças generalizadas;

$D(q)$ = matriz de inércia;

$V(q, \dot{q})$ = vetor de forças centrífugas e de Coriolis;

$G(q)$ = vetor de forças gravitacionais.

Os elementos da matriz $D(q)$ são dados por

$$D_{ik} = \sum_{j=\max(i,k)} Tr(U_{jk} J_j U_{ji}^T) \quad i, k = 1, \dots, n \quad (3.2)$$

com

$$U_{ij} \triangleq \frac{\partial^0 T_i}{\partial q_j} \quad (3.3)$$

que pode ser escrito como

$$U_{ij} = \begin{cases} {}^0T_{j-1} Q_j {}^{j-1}T_i & \text{para } j \leq i \\ 0 & \text{para } j > i \end{cases} \quad (3.4)$$

com

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{para juntas de revolução} \quad (3.5)$$

e

$$Q_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{para juntas prismáticas} \quad (3.6)$$

respeitadas as convenções de Denavit-Hartenberg (FU; GONZALEZ; LEE, 1987) para a atribuição dos sistemas de coordenadas.

A matriz de inércia do i -ésimo elo é definida como

$$J_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (3.7)$$

onde:

I_{jk} = tensores de inércia do elo i ;

m_i = massa do elo i ;

$\bar{x}_i, \bar{y}_i, \bar{z}_i$ = coordenadas do centro de gravidade do elo i .

É importante observar que, como a matriz de inércia $D(q)$ é simétrica, é necessário calcular através da expressão apenas os elementos para os quais $j \geq i$.

Os elementos do vetor de forças centrífugas e de Coriolis são dados por:

$$V_i = \sum_{k=1}^n \sum_{m=1}^n v_{ikm} \dot{q}_k \dot{q}_m \quad i = 1, \dots, n \quad (3.8)$$

sendo v_{ikm} dado por

$$v_{ikm} = \sum_{j=\max(i,k,m)}^n \text{Tr} (U_{jkm} J_j U_{ji}^T) \quad i, k, m = 1, \dots, n \quad (3.9)$$

com

$$U_{ijk} = \frac{\partial U_{ij}}{\partial q_k} \quad (3.10)$$

que pode ser escrito como

$$U_{ij} = \begin{cases} T_{j-1} Q_j T^{k-1} Q_k T_i & \text{para } i \geq k \geq j \\ T_{k-1} Q_k^{k-1} T_{j-1} Q_j T_i & \text{para } i \geq j \geq k \\ 0 & \text{para } i < j \text{ ou } i < k \end{cases} \quad (3.11)$$

Os elementos do vetor de forças gravitacionais são dados por

$$G_i = \sum_{j=1}^n (-m_j g^T U_{ji} {}^j\bar{r}_j) \quad i = 1, \dots, n \quad (3.12)$$

onde:

g = aceleração da gravidade;

${}^j\bar{r}_j$ = coordenadas do centro de gravidade do elo j em relação ao sistema de coordenadas j .

Convém notar que todos os vetores e matrizes utilizados nesta formulação devem estar representados no sistema de coordenadas da base do manipulador.

Definindo-se $x = [q^T \dot{q}^T]^T$ e $u = \tau$, o modelo (3.14) pode ser escrito no espaço de estados como

$$\dot{x} = \begin{bmatrix} 0 & I \\ -D^{-1}(q)G'(q) & -D^{-1}(q)V'(q, \dot{q}) \end{bmatrix} x + \begin{bmatrix} 0 \\ D^{-1}(q) \end{bmatrix} u \quad (3.13)$$

com $V'(q, \dot{q})\dot{q} = V(q)$ e $G'(q)q = G(q)$ e que pode ser escrita na seguinte forma:

$$\dot{x} = f(x) + g(x)u \quad (3.14)$$

3.2 Lei de controle

A lei de controle denominada Torque Calculado foi utilizada neste trabalho. O referido controlador utiliza as informações disponíveis sobre o modelo do robô (ASADA; SLOTINE, 1986), ou seja, utiliza estas informações na determinação efetiva do sinal de controle. Também vale destacar que foi realizada outra implementação, com lei de controle PID (ALT; LAGES, 2002) e concluiu-se que o Torque Calculado apresentava um melhor desempenho.

A lei de controle é composta por dois componentes: um em *feedforward* e outro em *feedback*, como pode ser visto na figura 3.1.

O componente em *feedforward* visa compensar os efeitos da gravidade e das forças de interação entre as várias juntas do robô. O componente em *feedback* produz as parcelas de torques necessárias para compensar os desvios em relação à trajetória nominal do manipulador. A lei de controle é então dada pela expressão (3.15)

$$\tau = D(q) [\ddot{q}_r + K_d(\dot{q}_r - \dot{q}) + K_p(q_r - q)] + V(q, \dot{q}) + G(q) \quad (3.15)$$

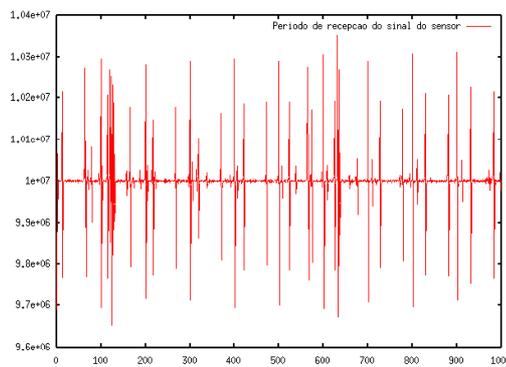
onde:

\ddot{q}_r = aceleração de referência;

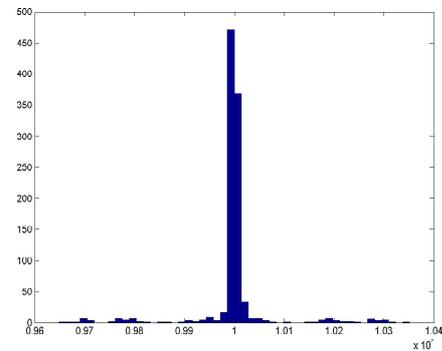
Os dados apresentados no sistema de controle local, portanto, são período e *jitter* de envio do sinal de controle, período e *jitter* de recepção do sinal do sensor e o tempo e *jitter* de *round-trip* que representaram as referências do modelo a serem atingidos com a rede inserida. Adicionalmente, apresentam-se também a média, o desvio-padrão e histograma das amostragens realizadas.

3.4 Resultados

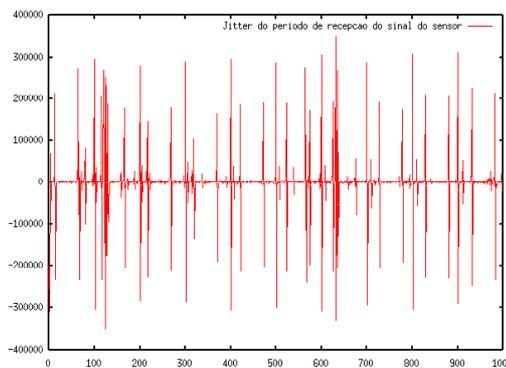
A figura 3.2 mostra os gráficos e os histogramas de período e do *jitter* de recepção do sinal dos sensores.



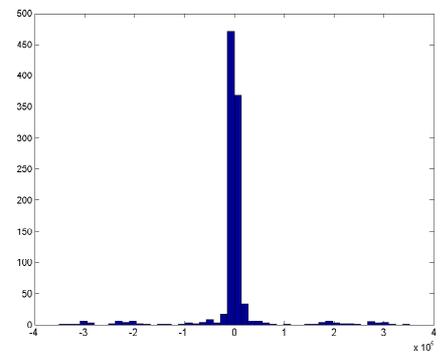
(a) Período (ns).



(b) Histograma do período.



(c) *Jitter* do período (ns).

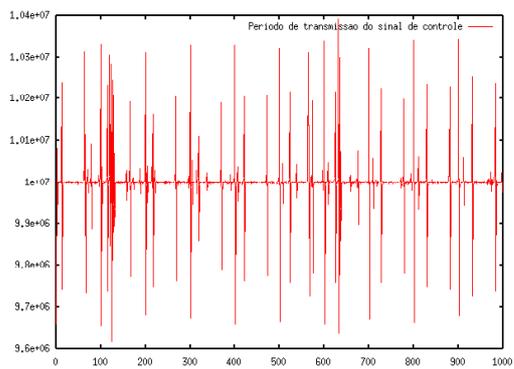


(d) Histograma do *jitter*.

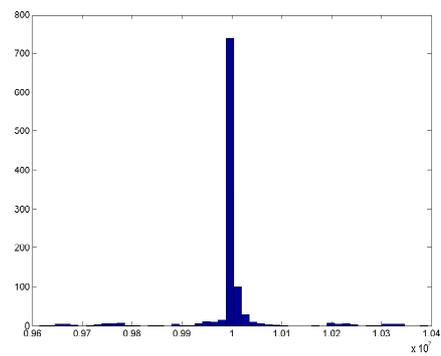
Figura 3.2: Período de recepção do sinal do sensor com controle local.

O valor de referência para a junta 1 é 45° e para a junta 2 é 90° .

O período de recepção do sinal de controle é o período de atualização dos atuadores. Com o controle e o manipulador robótico operando localmente, tipicamente este período será igual ao período de recepção do sinal dos sensores, conforme apresentado na figura 3.2. As pequenas diferenças devem-se às variações da velocidade de processamento do computador. A figura 3.3 apresenta os gráficos e os histogramas do período e do *jitter* de recepção do sinal de controle.



(a) Período (ns).



(b) Histograma do período.

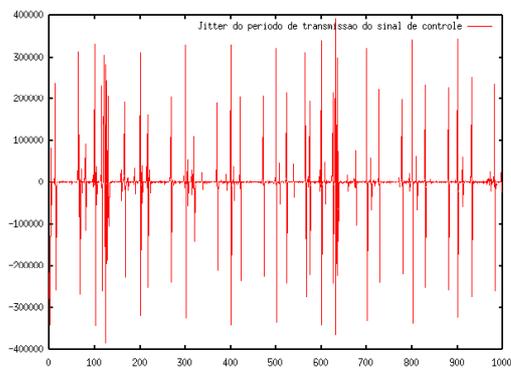
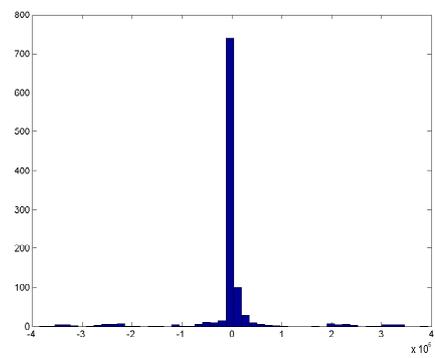
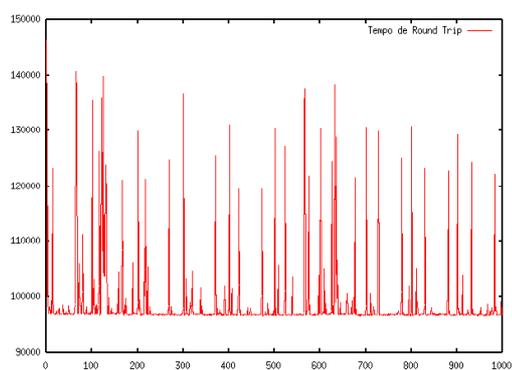
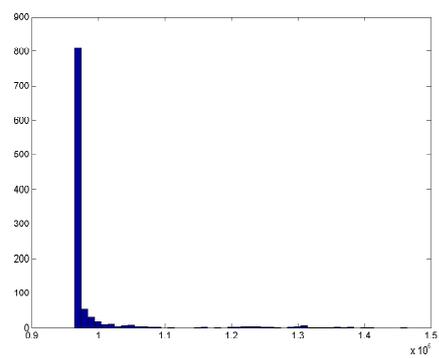
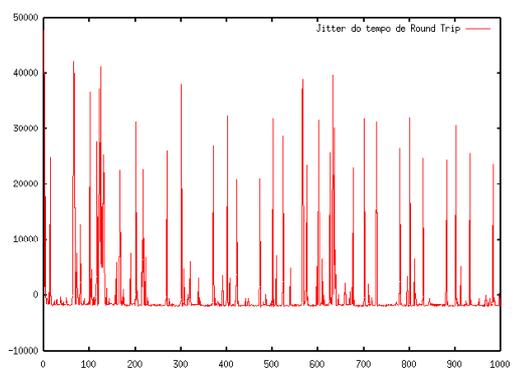
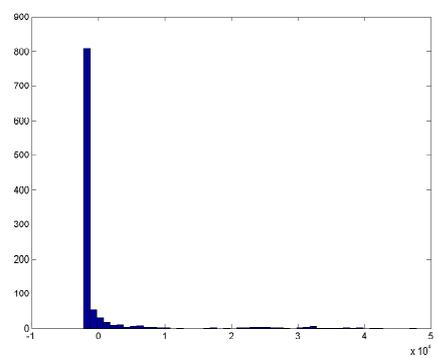
(c) *Jitter* do período (ns).(d) Histograma do *jitter* do período.

Figura 3.3: Período de envio do sinal de controle com controle local.

(a) Tempo de *round-trip* (ns).

(b) Histograma do período.

(c) *Jitter* do tempo de *round-trip* (ns).(d) Histograma do *jitter*.Figura 3.4: Tempo de *round-trip* com controle local.

O tempo de *round-trip*, neste caso, representa o tempo de computação da lei de controle. A figura 3.4 mostra os gráficos e os histogramas do tempo e do *jitter* de *round-trip*.

Como já mencionado, o desempenho obtido operando localmente é o ideal a ser obtido pelo sistema de controle através da rede. A tabela 3.1 mostra as médias e desvio padrão das métricas computadas para o controle local.

Tabela 3.1: Estatísticas das métricas com controle local.

Métrica	Média (segundos)	Desvio Padrão (segundos)
T_{rs}	$9999469,58 \times 10^{-09}$	$66084,92 \times 10^{-09}$
J_{rs}	$-530,42 \times 10^{-09}$	$66084,92 \times 10^{-09}$
T_{rc}	$9999419,75 \times 10^{-09}$	$73707,45 \times 10^{-09}$
J_{rc}	$-580,25 \times 10^{-09}$	$73707,45 \times 10^{-09}$
T_{rt}	$98621,90 \times 10^{-09}$	$6684,46 \times 10^{-09}$
J_{rt}	0	$6684,46 \times 10^{-09}$

O sistema apresenta a curva de resposta com controlador local indicada na figura 3.5. A tabela 3.2 mostra características de desempenho do sistema de controle. A posição angular de referência para a junta 1 é 45^0 e para a junta 2 é 90^0 .

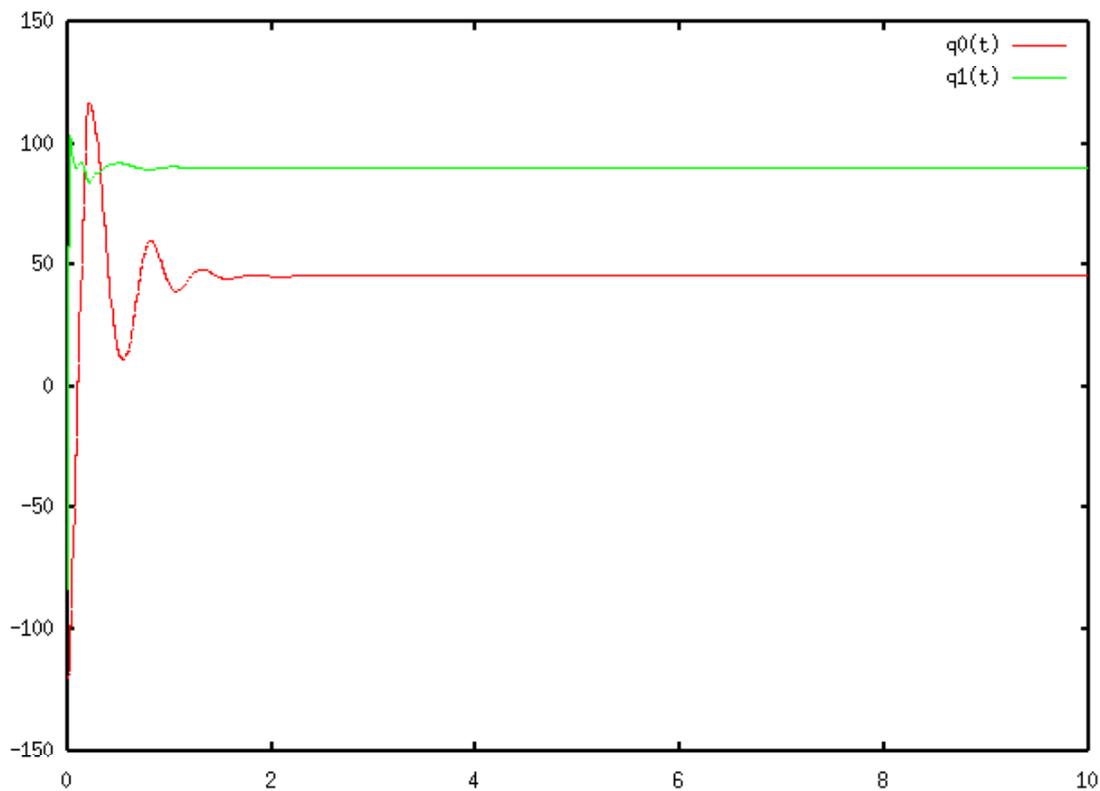


Figura 3.5: Resposta do controlador (posição angular em graus, tempo em segundos).

Tabela 3.2: Desempenho do sistema de controle com controle local.

	Junta 1	Junta 2
<i>Overshoot</i> (%)	43,63	6,19
Tempo de acomodação (segundos)	1,41	0,27
Tempo de subida (segundos)	0,135	0,018
Tempo de pico (segundos)	0,20	0,02
Critério ISE (radianos ²)	79,771579	11,442537
Critério IAE (radianos)	69,843623	11,692929
Critério ITSE (radianos ² × segundos)	10,999367	0,162832
Critério ITAE (radianos × segundos)	57,075757	23,881627

Os valores indicados através dos critérios ISE, IAE, ITSE e ITAE, apresentados na tabela 3.2, indicam o desempenho do sistema de controle; quanto menor o valor destes critérios, melhor é o desempenho do controle. Como mencionado na seção 3.2, está se utilizando uma lei de controle do tipo torque calculado. Os parâmetros do controlador poderiam ser sintonizados para um melhor desempenho, mas este não é o foco deste trabalho. A resposta do controlador, mostrada na figura 3.5, reflete o desempenho de um controlador que não está sujeito aos atrasos impostos pela rede. Embora este controle não seja o ideal, deseja-se que, com a introdução da rede na malha de controle, o desempenho seja mantido.

4 CONTROLE ATRAVÉS DE REDE IP

Neste capítulo, será apresentado o controle em rede sem qualquer técnica de compensação aplicada.

Este estudo tem por objetivo analisar o comportamento do controlador através da rede. Será apresentada a estrutura do controlador e do manipulador robótico operando em rede, representada na figura 4.1. Descreve-se também o protocolo de comunicação utilizado e o método para inserção de atrasos na rede IP.

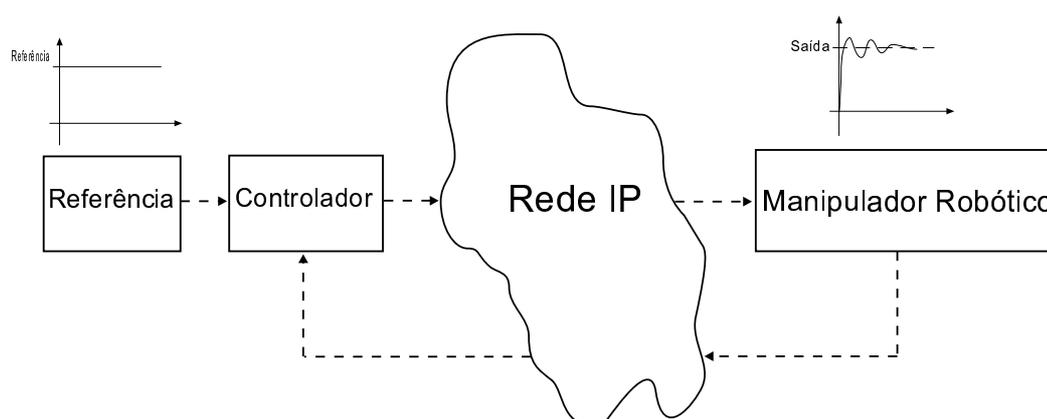


Figura 4.1: Estrutura do sistema de controle baseado em rede.

Nesta estrutura a rede IP é utilizada para enviar os sinais de controle e atuar entre os nós da rede. No manipulador robótico estão instaladas as interfaces que acionam os motores associados a cada uma das juntas do robô, assim como os respectivos sensores de cada junta; no controlador se realiza o cálculo de controle, baseado na lei de controle apresentada na seção 3.2. Os sinais de controle são transmitidos do controlador para o manipulador e os sinais dos sensores são transmitidos do manipulador para o controlador.

4.1 Descrição da rede

A rede está baseada no padrão Ethernet e nos protocolos UDP / IP. O IP permite endereçamento, garantindo a independência de cada datagrama. O IP pode fragmentar datagramas (em roteadores) e reagrupar estes. Com isso, é permitido se

trafegar em redes com características diferentes. Entretanto, a rede IP não permite assegurar que o dado seja entregue, roteadores podem descartar datagramas, sem avisar à origem ou ao destino. O IP confia nas camadas acima da camada de rede para gerenciar os quadros de datagramas e retransmití-los quando necessário. No entanto, este mecanismo pode somente assegurar a entrega de dados. Em nenhum momento, o protocolo IP ou mesmo as camadas de mais alto nível asseguram o tempo de entrega ou permitem qualquer garantia sobre a taxa de dados transmitidos. A camada IP oferece o que é chamado de serviço *best effort*, onde não se garante quando os dados chegarão ao seu destino, ou mesmo se serão entregues. Esta limitação não causa problema para aplicações como web, mail, transferência de arquivos. Mas, como as novas necessidades de aplicações, incluindo áudio e vídeo, demandam uma alta taxa de transferência de dados e necessitam de uma baixa latência quando utilizam comunicação de duas vias, existem atualmente inúmeras discussões e propostas para a incorporação de Qualidade de Serviço (XIAO; NI, 1999) sobre IP. Como melhorar o nível de serviços do protocolo Internet (IP) e a infra-estrutura sem adicionar complexidade, que poderá inviabilizar a rede, tem se inserido no contexto destas discussões.

A figura 4.2 apresenta as camadas de rede com os diversos protocolos utilizados neste trabalho.

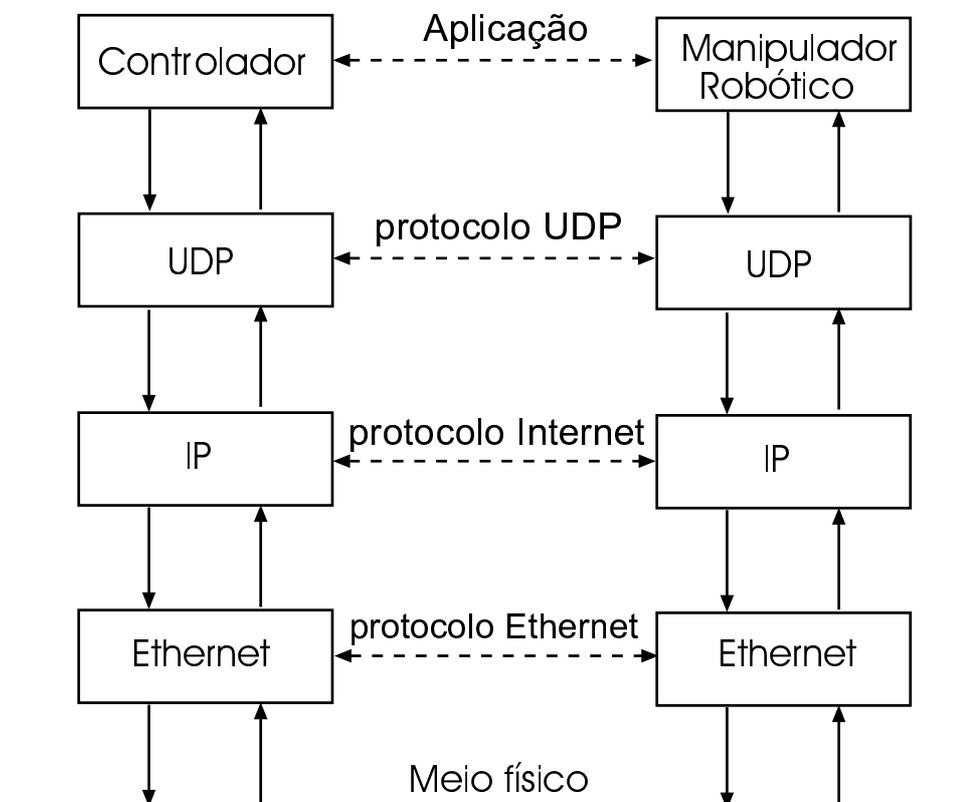


Figura 4.2: Protocolos de rede utilizados para o controle do robô através da rede.

4.1.1 Rede Ethernet

Neste trabalho, optou-se por utilizar, para os níveis de enlace e físico, o protocolo Ethernet, que é um protocolo não determinístico. Esta escolha deve-se à sua grande disponibilidade e seu custo reduzido porque atualmente está muito difundida no mercado. Importante notar que o protocolo Ethernet apresenta um desempenho superior ao apresentado pelos protocolos dedicados a redes de controle, como Profibus, Fieldbus, ControlNet e CAN (LIAN; MOYNE; TILBURY, 2001), quando a rede está pouco carregada. No entanto, as suas principais desvantagens estão relacionadas com o controle de acesso ao meio através de CSMA/CD (*Carrier Sense Multiple Access*), que torna o tempo de transmissão dos pacotes pouco previsível; portanto, adequada para utilização em redes de controle.

O uso de equipamentos como *switches* reduz bastante o problema, mas ainda assim persistirão algumas colisões; por exemplo, quando diversos pacotes endereçados a mesma porta chegam ao *switch* simultaneamente.

Muito dos trabalhos recentes na área, que suportam comunicações em tempo real através da Ethernet, focam na modificação das sub-camadas da MAC da Ethernet, de modo que um limite do tempo de acesso ao meio possa ser obtido (CHOI et al., 2000). Estas modificações são muito caras comparado a utilização do protocolo estabelecido e largamente utilizado atualmente no padrão Ethernet. Existem trabalhos de alteração da camada MAC para evitar a colisão de pacotes, denominado de *token ring* virtual (KWEON; SHIN, 2003). Entretanto, estas modificações causam grandes transtornos, tendo em vista a alteração necessária do protocolo MAC, inclusive podendo degradar o performance do *kernel* do sistema operacional, devido à complexidade do gerenciamento do *token*.

Neste trabalho, não se tem por objetivo realizar alterações do protocolo MAC. O objetivo será a utilização de técnicas de compensação dos atrasos na camada da aplicação.

4.1.2 Protocolos UDP

UDP é um protocolo de datagramas não orientado a conexão. Alguns exemplos de aplicações populares construídas utilizando UDP são: DNS (*Domain Name System*), NFS (*Network File System*), e SMNP (*Simple Network Management Protocol*), por exemplo.

Em princípio, a utilização do protocolo TCP parece ser mais conveniente, pois garante uma comunicação sem perda de mensagem. No entanto, o protocolo TCP não possui mecanismos para garantir a temporização das mensagens. Em uma rede de controle, mensagens com atrasos de rede muito grande são inúteis, pois os dados de sensores ou controle estarão desatualizados. Aquelas mensagens que são retransmitidas acarretam uma ocupação adicional da largura de banda do sistema para transmitir uma mensagem que não mais será aproveitada, pois a mesma será descartada por atraso. Em casos limites, pode-se chegar à situação de congestionamento da rede, onde as retransmissões fazem com que a largura de banda seja excedida, provocando descarte de mensagens que terão que ser retransmitidas e assim por diante.

Por outro lado, em um sistema de controle digital, tipicamente, os sinal dos sen-

sores são amostrados a uma taxa fixa, de forma que as mensagens com a leitura dos sinais dos sensores sejam transmitidas regularmente. No caso do sistema de controle desenvolvido, utiliza-se uma taxa de 100 Hz. Se a taxa de perda das mensagens da rede não for muito grande, pode ser mais conveniente aguardar pela próxima mensagem com a leitura dos sensores do que retransmitir uma mensagem perdida para obter-se uma leitura desatualizada. Pode-se também utilizar leituras anteriores do sinal do sensor para estimar a leitura correspondente às mensagens perdidas, considerando o erro entre a estimativa e a leitura perdida como se fosse parte do erro inerente ao sinal do sensor. Com estas considerações, torna-se natural a utilização do protocolo UDP e tem-se o benefício adicional de utilizar menos largura de banda que o protocolo TCP. Note-se, no entanto, que o protocolo UDP também não garante a temporização das mensagens.

4.1.3 Leitura do datagrama

A comunicação entre o robô e o controlador é implementada através da utilização de *sockets* UDP. Duas portas de transmissão são utilizadas para criar dois canais virtuais no sistema. Um canal é para a transmissão do sinal do sensor e o outro canal é para os dados amostrados pelos sensores.

Para a leitura do datagrama, foi escrita uma função que permanentemente está em laço infinito lendo o próximo datagrama na porta do servidor. Na figura 4.3 mostra-se o esquema como o controlador envia datagramas para o manipulador robótico utilizando UDP.

Os principais atrasos de transporte em sistemas de controle distribuídos não são físicos, pois a velocidade da rede é tal que o tempo de transmissão de dados é muito pequeno. No sistema amostrado convencional se assume que o instante de tempo da transmissão do sinal do sensor ($t_{ts}(k)$) pelo manipulador robótico e o instante de tempo da recepção do sinal do sensor ($t_{rs}(k)$) pelo controlador são o mesmo instante, o que não é verdadeiro para sistemas de controle distribuído. Adicionalmente, devido aos atrasos da rede, o controlador pode não receber todas as atualizações da saída do manipulador robótico no tempo previsto para o cálculo do controle dentro do período de amostragem.

Pacotes são descartados, ocasionalmente, no sistema de controle distribuído, quando existem falhas no nó quando há colisão de mensagens ou quando o *buffer* esgotou sua capacidade. Mesmo que o protocolo tenha mecanismos de retransmissão, ele somente pode realiza-las por um tempo limitado. Depois que este tempo expira, os pacotes são descartados. Além disso, para dados de controle em tempo real como as medidas dos sensores e cálculo dos sinais de controle, pode ser vantajoso não transmitir dados antigos e sim transmitir um novo pacote. Desta maneira, o controlador sempre recebe um novo dado para calcular o controle.

4.2 Manipulador robótico - daemon robô

O *software* executado no robô tem por função realizar a interface com o robô, permitindo o acionamento das juntas e leitura dos seus sensores. O programa está dividido em três *threads* independentes, que são:

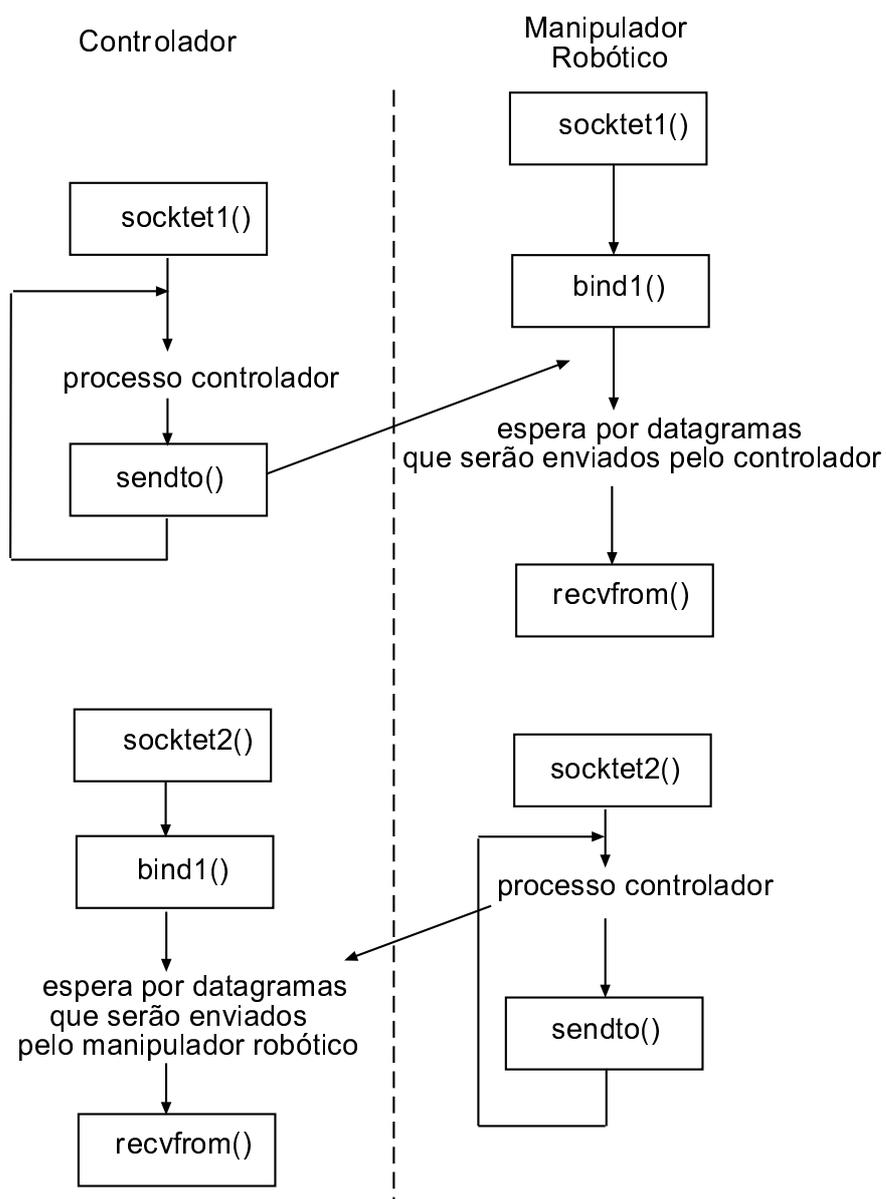


Figura 4.3: Comunicação controlador / robô através do protocolo UDP.

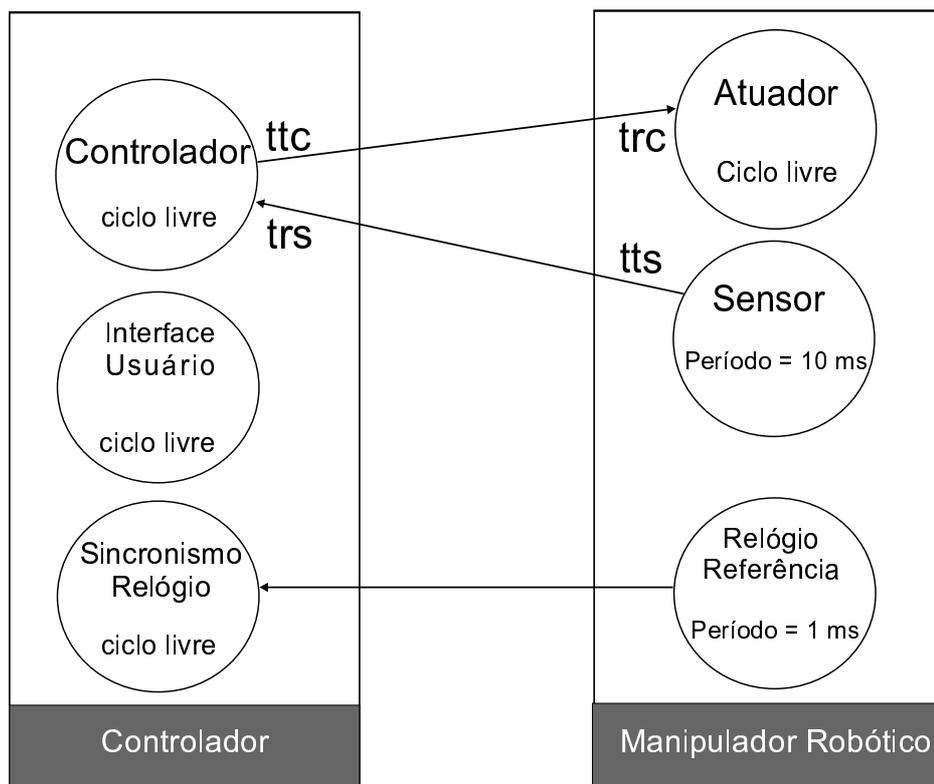


Figura 4.4: Esquema utilizando manipulador robótico temporizado.

Atuador: interface de recepção com o controlador e atuador do robô. Recebe o sinal de controle e aplica na respectiva junta do robô, executando os comandos enviados pelo controlador remoto. Seu ciclo de operação é livre e executado de forma seqüencial, coordenado pela operação do controle. É implicitamente temporizado pela recepção dos pacotes do controlador.

Sensor: interface com os sensores do robô e tem por função transmitir os sinais dos sensores. A transmissão é executada de forma periódica com um período fixo de 10 ms (100 Hz). A transmissão independe das demais *threads*.

Relógio Referência: tem por finalidade ser o relógio de referência do sistema de controle e manipulador robótico. A frequência da *thread* do relógio é de 1 KHz.

O fluxo seguido pelos sinais de controle e do manipulador robótico é representado na figura 4.4. Ao ser inicializado, o manipulador robótico pelo controlador, a primeira ação é transmitir os sinais dos sensores (no instante t_{ts}), que são recebidos no programa controlador sensores (no instante t_{rs}). O controlador processa o cálculo do controle e transmite o sinal de controle para atuador (no instante t_{tc}), que por sua vez, é recebido pelo manipulador robótico (no instante t_{rc}).

Para se aplicar a técnica de compensação, foi implementado um protocolo de sincronização do relógio entre as estações que compõem o sistema do robô. As funções de tempo padrão do sistema Linux, ou do RTAI, não podem ser utilizadas para o *time-stamp* dos pacotes, porque o tempo retornado por elas não é mantido

com a necessária sincronização entre os nodos da rede. Mesmo com a utilização de *daemons* de sincronização que implementam os protocolos de sincronismo de tempo, como o NTP, a sincronização não foi suficiente para medir o atraso da rede.

Para resolver o problema, foi desenvolvido o protocolo que fornece uma referência de tempo para o sistema de controle distribuído apresentado neste trabalho. Este relógio de referência permite a operação simples e estabelece uma resolução de tempo de 1 ms, e é executado por uma tarefa independente executada em cada um dos nodos (vide figura 4.4). O protocolo desenvolvido para sincronizar o relógio foi implementado para assegurar que os pacotes possam trafegar entre a rede com a marcação de tempo precisa e padronizada no sistema, permitindo usar as técnicas de estimação propostas. A implementação deste protocolo foi realizada através da utilização da porta serial dos computadores envolvidos. Para computadores em pontos físicos distintos, pode-se utilizar outro método de sincronização, como por exemplo, sincronização do relógio através do GPS; ou até mesmo NTP, caso o sistema seja menos exigente do que um robô.

4.3 Controlador

Este programa tem por função controlar através da rede o sistema robótico, segundo a lei de controle apresentada na seção 3.2. O programa é formado por três *threads* independentes, que são:

Controlador: recebe o sinal dos sensores do robô, calcula o controle, segundo a lei de controle do tipo torque calculado (CRAIG, 1986; FU; GONZALEZ; LEE, 1987; SPONG; VIDYASAGAR, 1989) e transmite o sinal de controle destas juntas. Seu ciclo de operação é livre, ou seja, sem temporização forçada. A periodicidade de recepção do sinal de controle é quem dita a frequência de operação do controlador.

Sincronismo de Relógio: Sincroniza o relógio de controle segundo a referência determinada pelo manipulador robótico. A frequência de execução da *thread* de sincronismo do relógio é livre, em função da operação do manipulador robótico.

Interface com o Usuário: tem por finalidade realizar a interface entre o sistema de controle e o usuário.

O controlador é orientado a eventos. O sinal do controlador é calculado assim que o sinal do sensor seja recebido do manipulador robótico. Realizado o cálculo, o controlador transmite o novo sinal de controle ao manipulador.

Quando a interface do usuário é acionada para desligar o controlador, esta transmite o sinal de desligar ao robô, o qual finaliza suas operações, interrompendo a transmissão do sinal de controle.

Os dados transmitidos do controlador para o manipulador robótico e vice-versa, utilizam um protocolo de aplicação (vide figura 4.2), especialmente desenvolvido.

4.4 Protocolo de Aplicação

O pacote genérico transmitido entre os programas de aplicação utiliza uma *string* e é apresentado na figura 4.5. A tabela 4.1 detalha os campos deste pacote, que é transmitido através de UDP.



Figura 4.5: Formato do pacote genérico de controle.

Tabela 4.1: Descrição do pacote genérico de controle.

Descrição	
cmd	operação a ser realizada pelo controlador ou manipulador robótico: LIGA = 'L' DESLIGA = 'D' CONTROLE = 'C' LEITURA DOS SENSORES = 'S'
argumentos	utilizado pelo controle.
<i>time stamp</i>	instante de tempo de envio do pacote.
CRC	campo destinado para conferência, calculado pelo polinômio $X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$.

Os campos são transmitidos em formato ASCII, separados por espaço. Os dados, que formam o campo destinado a argumentos, são de tamanho variáveis em função do número de juntas a serem controladas. Também é importante ressaltar que alguns dos campos não são utilizados em função do tipo de pacote transmitido. O CRC utiliza o mesmo polinômio utilizado pelo barramento CAN.

4.4.1 Pacotes do controlador

1. Pacote de Controle

O pacote transmitido do controlador remoto para o manipulador robótico, com o sinal de controle, é apresentado na figura 4.6 e descrito na tabela 4.2.

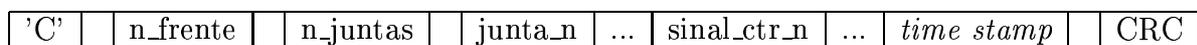


Figura 4.6: Formato do pacote de controle.

Tabela 4.2: Descrição do pacote de controle.

	Descrição
'C'	especificação da operação a ser realizada: atuação das juntas
n_frente	quantidade de passos de controle enviados.
quant_juntas	quantidade de juntas comandadas.
junta_n	especificação da junta n a ser comandada. A quantidade de especificações de juntas é variável em função do número de juntas comandadas.
sinal_ctr_n	sinal de controle da junta n. A quantidade dos dados será variável em função do número de juntas a serem comandadas.
<i>time stamp</i>	instante de tempo de envio do pacote.
CRC	campo destinado para conferência do pacote, calculado pelo polinômio $X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$.

Os dados, que fazem parte do campo destinado a argumento, são variáveis em função do número de juntas a serem controladas.

2. Pacote LIGA

O pacote transmitido pelo controlador, para ativar o manipulador robótico, é apresentado na figura 4.7 e detalhado na tabela 4.3.

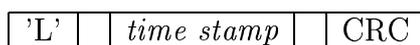


Figura 4.7: Formato do pacote Liga.

Tabela 4.3: Descrição do pacote Liga.

	Descrição
'L'	especificação da operação a ser realizada: Liga
<i>time stamp</i>	instante de tempo de envio do pacote liga.
CRC	campo destinado para conferência do pacote, calculado pelo polinômio $X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$.

3. Pacote DESLIGA

O pacote transmitido pelo controlador, para desativar o manipulador robótico, é apresentado na figura 4.8 e detalhado na tabela 4.4.

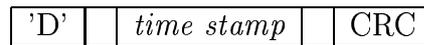


Figura 4.8: Formato do pacote Desliga.

Tabela 4.4: Descrição do pacote Desliga.

Descrição	
'D'	especificação da operação a ser realizada: Desliga
<i>time stamp</i>	instante de tempo de envio do pacote desliga.
CRC	campo destinado para conferência do pacote calculado pelo polinômio $X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$.

4.4.2 Pacote transmitido pelo manipulador robótico

O pacote transmitido do manipulador robótico para o controlador é apresentado na figura 4.9 e detalhado na tabela 4.5.

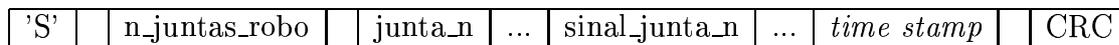


Figura 4.9: Formato do pacote do manipulador robótico.

Tabela 4.5: Descrição do pacote de leitura dos sensores.

Descrição	
'S'	especificação do tip de operação realizada: Leitura dos Sensores
junta_n	especificação da junta n. A quantidade de especificações de juntas será variável em função do número de juntas a serem acionadas.
sinal_junta_n	dado do sinal de sensor da junta n do manipulador robótico. A quantidade dos dados será em função do número de juntas a serem acionadas.
<i>time stamp</i>	instante de tempo de envio do pacote do manipulador robótico.
CRC	campo destinado para conferência do pacote calculado pelo polinômio $X^{15} + X^{14} + X^{10} + X^8 + X^7 + X^4 + X^3 + 1$.

Os dados que são contemplados no campo destinado a argumento são variáveis em função do número de juntas a serem controladas.

4.5 Inserção de atrasos

Para impor atrasos conhecidos aos pacotes entre o controlador remoto e o sistema robótico, foi desenvolvido um programa que atua junto ao roteador da rede (ALT; LAGES, 2003b), permitindo emular o atraso da rede através da análise dos cabeçalhos IP e inserir atrasos distintos aos pacotes com sinal de controle e com sinal do sensor, simulando as condições de carga na rede (vide figura 4.10).

O emulador de tráfego é baseado nas facilidades do Netfilter (NETFILTER, 2003) incluído no *kernel* do Linux. Conjuntamente com IPtables, o Netfilter pode inserir os pacotes selecionados que trafegam através da rede em uma fila, a ser processada por programas que são executados no espaço do usuário. O Netfilter permite que seja determinado, pelo usuário, quais pacotes que passam pelo roteador e serão aceitos e seguirão adiante ou serão descartados.

O emulador de tráfego impõe atrasos aos pacotes selecionados. Estes atrasos são gerados de acordo com um modelo estatístico da carga de rede.

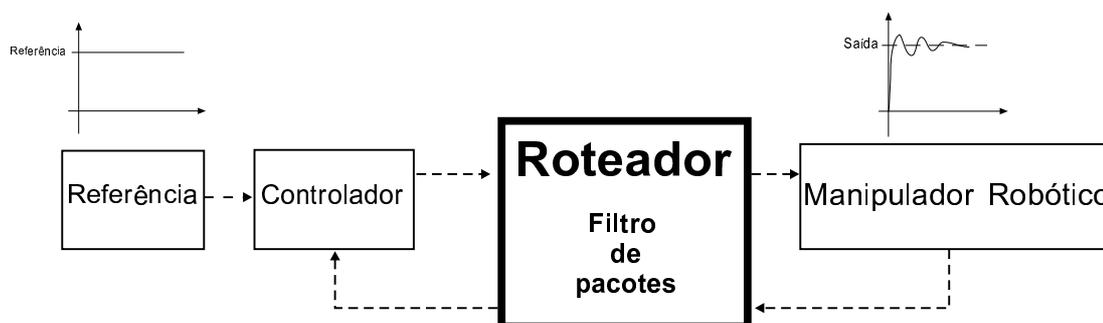


Figura 4.10: Filtro para inserção de atrasos.

O emulador de tráfego é um programa baseado em tarefas LXRT (vide seção 2.2), e processa os pacotes enfileirados pelo Netfilter. Este utilitário é formado por duas tarefas LXRT e por um processo do Linux.

O controle do fluxo de pacotes no roteador está constituído de um programa principal com duas *threads*:

escalonador: adiciona os pacotes em uma lista ligada ordenada por instante de liberação do pacote. Esta tarefa de LXRT tem por função agendar o pacote com o tempo para futura liberação. O atraso dos pacotes é determinado segundo o modelo estatístico e dado pela expressão (4.1), que está diretamente vinculado ao endereço de destino do pacote.

dispatcher: tem a função de realizar a liberação do pacote. Para executar a liberação dos pacotes da lista, é analisado o tempo em que cada pacote deve ser liberado da fila. Os pacotes que possuem o tempo da liberação expirado são removidos da lista e conduzidos ao subsistema da rede. Após ter despachado todos os pacotes com tempo expirado, o *dispatcher* fica em repouso até o momento da nova liberação do primeiro pacote que permaneceu na lista.

Esta arquitetura do escalonador/*dispatcher* é conveniente devido ao modelo estatístico de atraso imposto aos pacotes.

Os atrasos foram modelados pela combinação de duas distribuições, uma representando o tráfego de baixa carga e outra representando o tráfego de alta carga. Estas duas distribuições são combinadas através de uma distribuição uniforme, segundo a expressão (4.1).

$$D(t) = \begin{cases} D_H(t) & \text{se } P_{HL}(t) < p_{HL} \\ D_L(t) & \text{caso contrário} \end{cases} \quad (4.1)$$

onde:

$D(t)$ = atraso imposto ao pacote;

$D_H \sim N(d_H, \sigma_H)$ = representa o tráfego de alta carga;

$D_L \sim N(d_L, \sigma_L)$ = representa o tráfego de baixa carga;

$P_{HL} \sim U(0, 1)$ = representa a probabilidade de ter-se um dos dois tipos de tráfego;

p_{HL} = proporção entre os dois tipos de tráfego;

sendo $N(d, \sigma)$ uma distribuição normal com média d e desvio padrão σ e $U(u_0, u_1)$ uma distribuição uniforme entre u_0 e u_1 .

Note que D_H e D_L são truncadas para gerarem apenas valores positivos.

4.6 Resultados

Na sequência, são apresentados os tempos medidos com os respectivos gráficos de cada um dos experimentos realizados.

O valor de referência para a junta 1 é 45^0 e para a junta 2 é 90^0 .

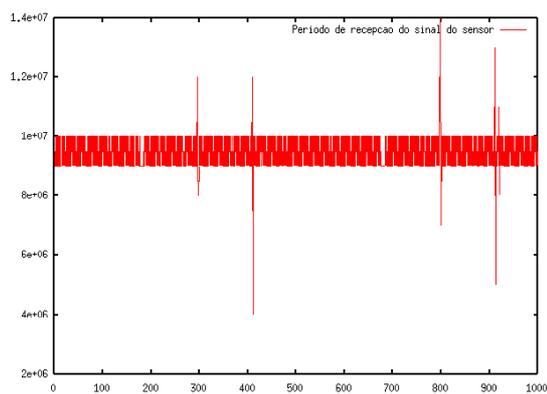
4.6.1 Controle através da rede sem atraso

Neste caso, apresenta-se o manipulador robótico e controlador, realizando a comunicação através da rede sem qualquer atraso introduzido e sem compensação de atraso. Este estudo tem por objetivo analisar os tempos dos pacotes na rede e verificar o desempenho do sistema de controle.

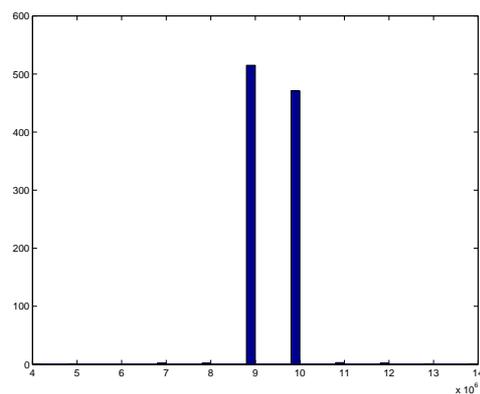
O período de amostragem dos sensores é basicamente constante, uma vez que existem apenas variações do sistema de tempo real. O período de recepção do sinal dos sensores não varia, devido a atrasos na comunicação no sentido robô/controlador serem desprezíveis.

O período de recepção do sinal de controle é o período de atualização dos atuadores. O período obtido neste experimento sofre pequenas variações devido à operação em rede, pois não há inserção de atrasos na comunicação nos sentidos manipulador robótico para o controlador e do controlador para o manipulador robótico. A figura 4.12 mostra os gráficos e os histogramas do período e do *jitter* de recepção do sinal de controle.

O tempo de *round-trip* também sofre pequenas influências do sistema operando em rede, mesmo não havendo inserção de atrasos no sentido do manipulador robótico



(a) Período (ns).



(b) Histograma do período.

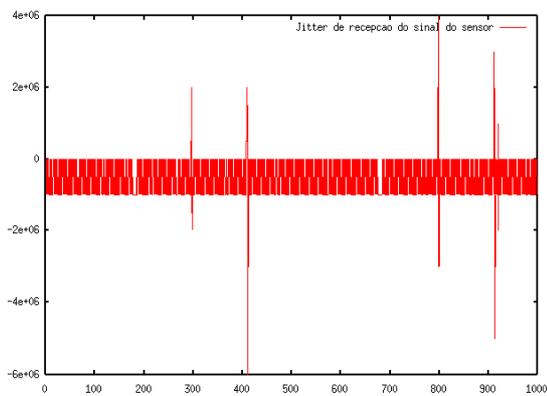
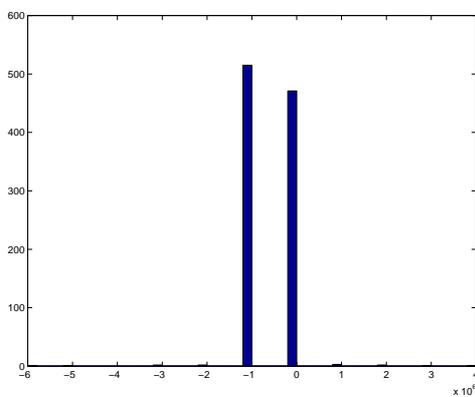
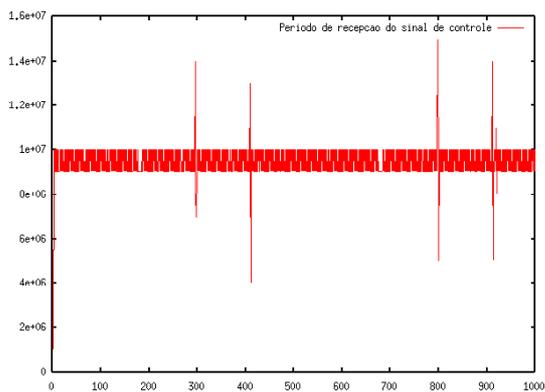
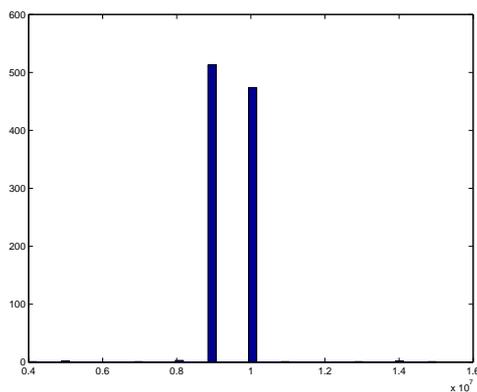
(c) *Jitter* do período (ns).(d) Histograma do *jitter* do período.

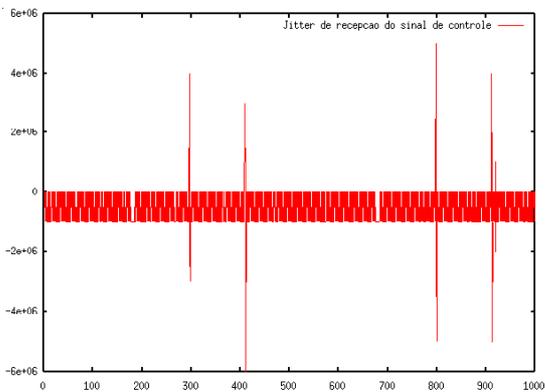
Figura 4.11: Período de recepção do sinal de sensor com controle através da rede sem atraso.



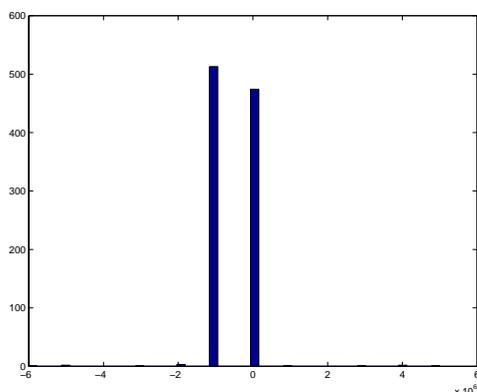
(a) Período (ns).



(b) Histograma do período.



(c) *Jitter* do período (ns).



(d) Histograma do *jitter* do período.

Figura 4.12: Período de recepção do sinal de controle com controle através da rede sem atraso.

para o controlador e do controlador para o manipulador robótico. Percebe-se variações na operação do sistema em relação a um sistema de controle convencional, sem rede. A figura 4.13 mostra os gráficos e os histogramas do tempo e do *jitter* de *round-trip*.

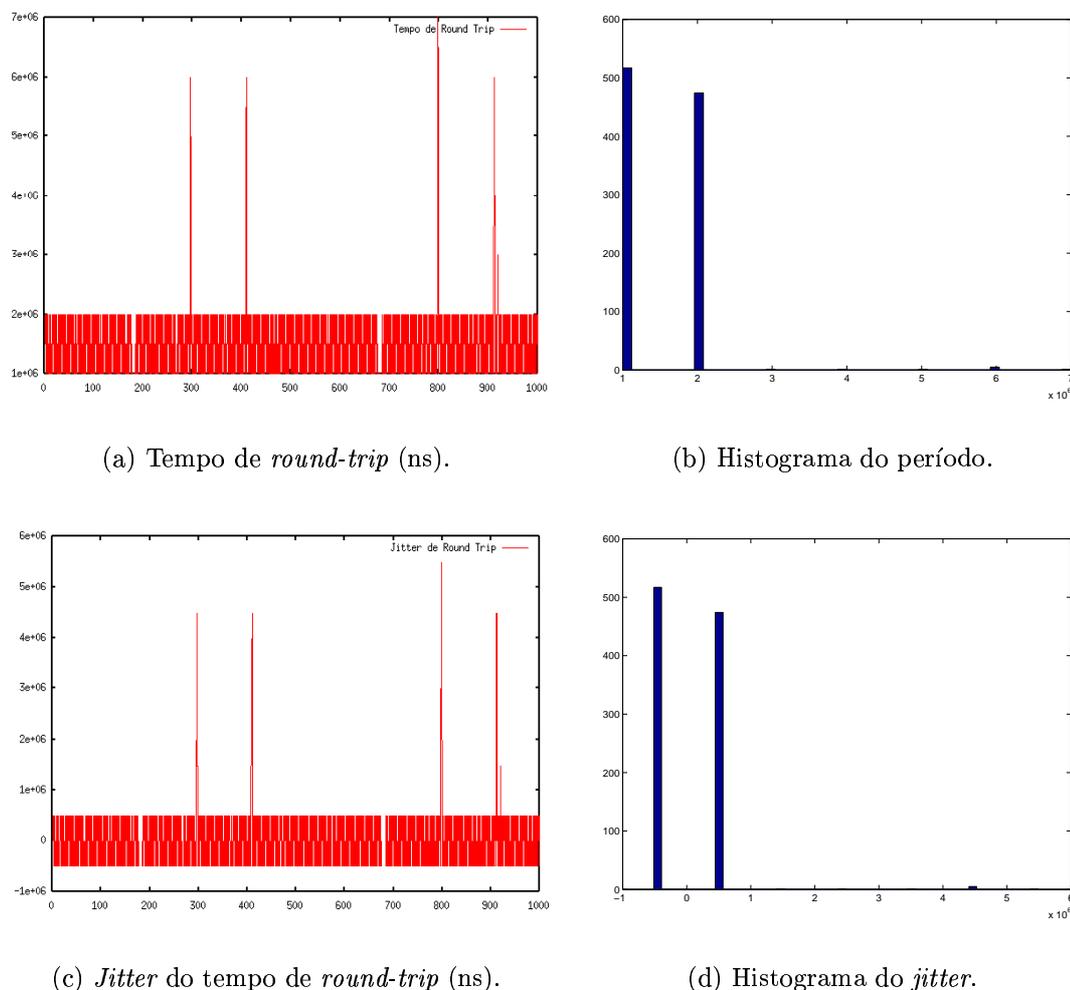


Figura 4.13: Tempo de *round-trip* com controlador com controle através da rede sem atraso.

A tabela 4.6 mostra os indicadores de temporização medidos através da rede IP sem inserção de atraso. Percebe-se uma variação dos tempos tomados no experimento, ficando o sistema de controle mais susceptível aos atrasos, o que resulta em operação do sistema de controle dentro do tempo de amostragem de 10 ms.

Em comparação com os resultados obtidos no controle local (tabela 3.1), nota-se que o comportamento da temporização já indica uma variação nos seus limites de operação. Há uma variação da temporização do sistema de controle através da rede, em comparação ao sistema local convencional, apesar do desempenho do sistema não ser afetado.

A figura 4.14 mostra as curvas de resposta do controle com a utilização do controlador através da rede sem atraso. A posição angular de referência para a junta 1

Tabela 4.6: Estatísticas das métricas com controle através da rede sem atraso.

Métrica	Média (segundos)	Desvio Padrão (segundos)
T_{rs}	$9477955,91 \times 10^{-09}$	$604769,91 \times 10^{-09}$
J_{rs}	$-522044,09 \times 10^{-09}$	$604769,81 \times 10^{-09}$
T_{rc}	$9478478,48 \times 10^{-09}$	$644750,33 \times 10^{-09}$
J_{rc}	$-521521,52 \times 10^{-09}$	$644750,33 \times 10^{-09}$
T_{rt}	1514000×10^{-09}	$632300,56 \times 10^{-09}$
J_{rt}	0	$632300,56 \times 10^{-09}$

é 45° e para a junta 2 é 90° . A tabela 4.7 mostra algumas características de desempenho do sistema de controle em rede sem atraso. Nota-se que o comportamento é muito similar ao verificado no sistema com controle local (tabela 3.2), como seria esperado.

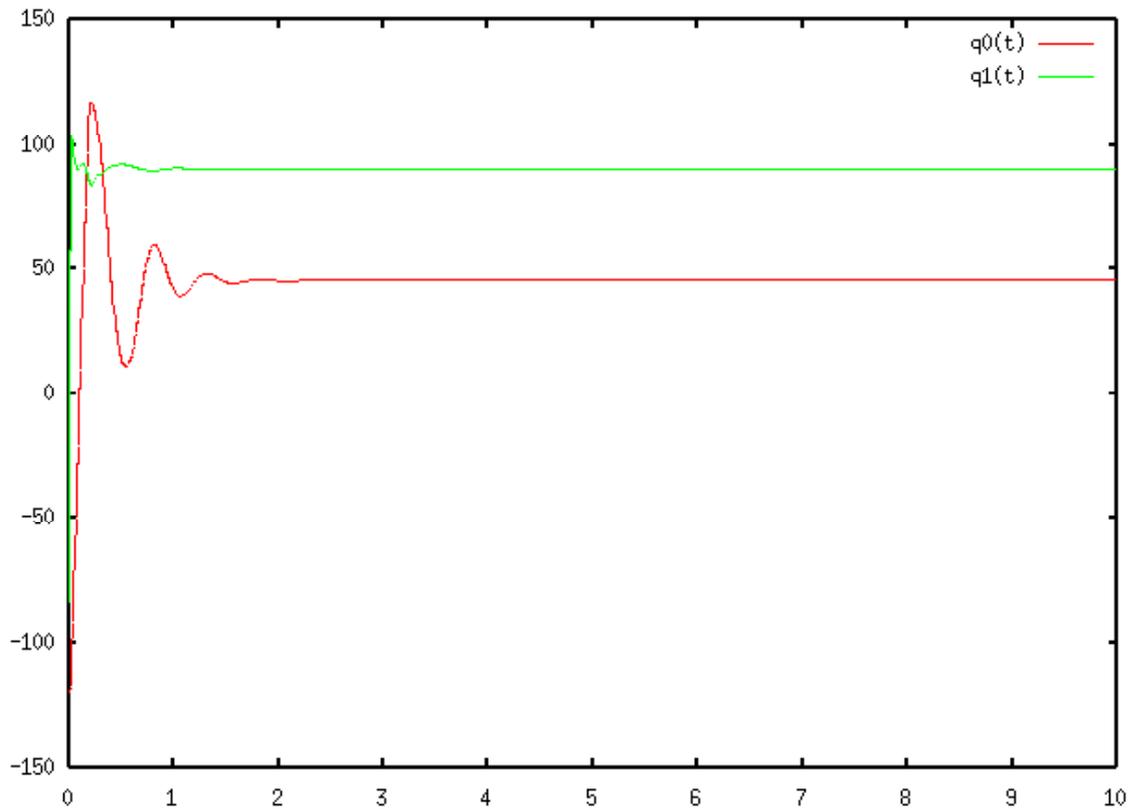


Figura 4.14: Resposta do controlador (posição angular em graus, tempo em segundos).

A partir da análise dos resultados acima, pode-se perceber que, na ausência de atrasos excessivos, o controle via rede é viável.

Tabela 4.7: Desempenho do sistema de controle através da rede sem atraso.

	Junta 1	Junta 2
<i>Overshoot</i> (%)	43,36	6,43
Tempo de acomodação (segundos)	1,41	0,27
Tempo de subida (segundos)	0,135	0,02
Tempo de pico (segundos)	0,20	0,02
Critério ISE (radianos ²)	79,781163	11,442553
Critério IAE (radianos)	70,345356	11,690898
Critério ITSE (radianos ² × segundos)	11,065322	0,162825
Critério ITAE (radianos × segundos)	61,041562	23,857616

4.6.2 Controle através da rede com atraso

Nesta situação, o controlador e o manipulador robótico comunicam-se através da rede com atraso inserido. Assim, embora o período de amostragem dos sensores seja basicamente constante, variando apenas devido a *jitters* introduzidos pelo sistema de tempo real, o período de recepção do sinal dos sensores varia devido ao atraso imposto pela rede. Este período sofre variações devido aos atrasos introduzidos na comunicação nos dois sentidos (no sentido manipulador robótico para controlador, como vice-versa). A figura 4.15 mostra os gráficos, com os respectivos histogramas, do período e do *jitter* de recepção do sinal dos sensores.

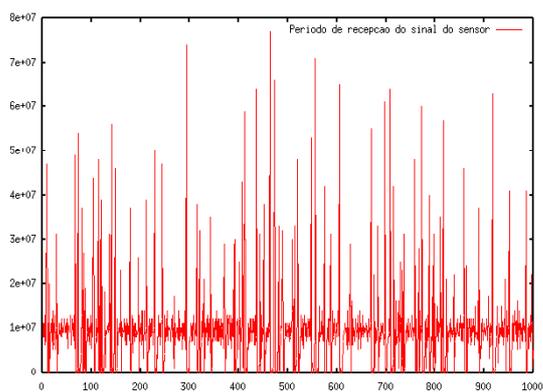
O período de recepção do sinal de controle é o período de atualização dos atuadores. Com o controle e o robô operando em rede, este período sofre variações devido aos atrasos na comunicação no sentido manipulador robótico para controlador e no sentido controlador para manipulador robótico. A figura 4.16 mostra os gráficos, e os respectivos histogramas, do período e do *jitter* de recepção do sinal de controle.

O tempo de *round-trip* também sofre influência dos atrasos introduzidos no sentido manipulador robótico para controlador e no sentido controlador para manipulador robótico. A figura 4.17 mostra os gráficos e os histogramas do tempo e do *jitter* de *round-trip*, onde percebe-se os atrasos significativos no período para transmitir o sinal de controle causados pelos atrasos de rede, e o conseqüente distúrbio causado na resposta de controle apresentado na figura 4.18.

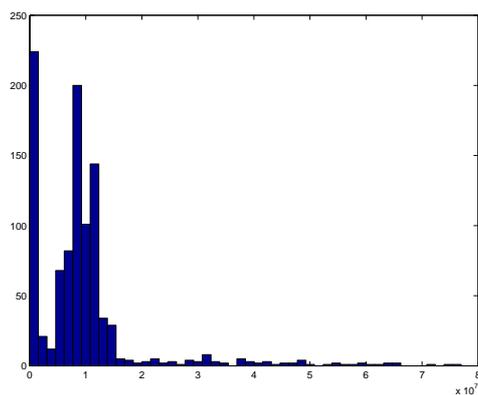
Tabela 4.8: Estatísticas das métricas com controle através da rede com atraso.

Métrica	Média (segundos)	Desvio Padrão (segundos)
T_{rs}	9469469, 47×10^{-09}	10413417, 21×10^{-09}
J_{rs}	-530530, 53×10^{-09}	10413417, 21×10^{-09}
T_{rc}	9467467, 47×10^{-09}	12626405, 62×10^{-09}
J_{rc}	-532532, 53×10^{-09}	12626405, 62×10^{-09}
T_{rt}	14745000×10^{-09}	15197498, 97×10^{-09}
J_{rt}	0	15197498, 97×10^{-09}

Em comparação ao sistema de controle através da rede sem atraso (tabela 4.6) e, mesmo em comparação com as métricas de rede do sistema de controle local



(a) Período (ns).



(b) Histograma do período.

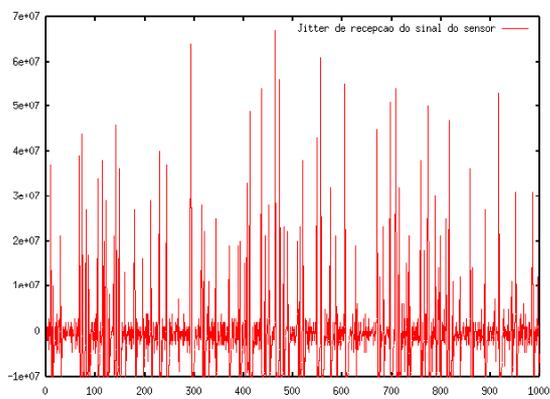
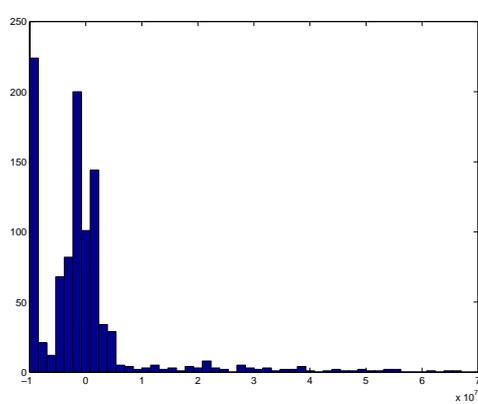
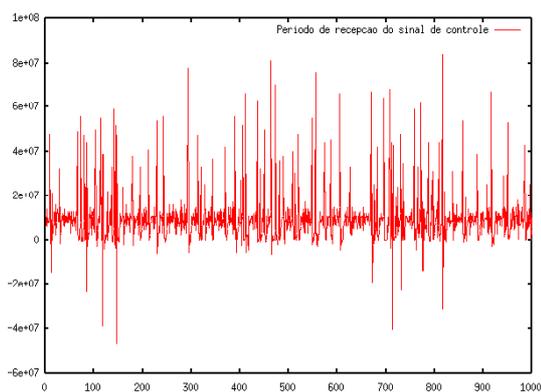
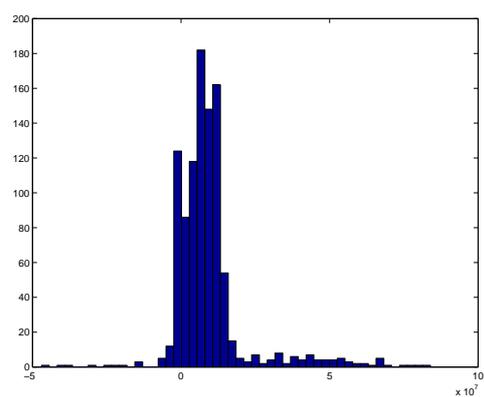
(c) *Jitter* do período (ns).(d) Histograma do *jitter* do período.

Figura 4.15: Período de recepção do sinal de sensor com controle através da rede com atraso.



(a) Período (ns).



(b) Histograma do período.

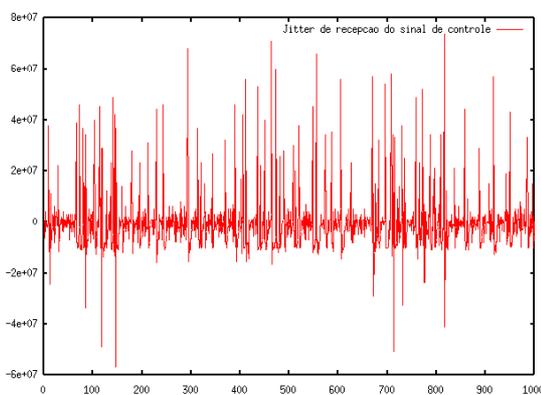
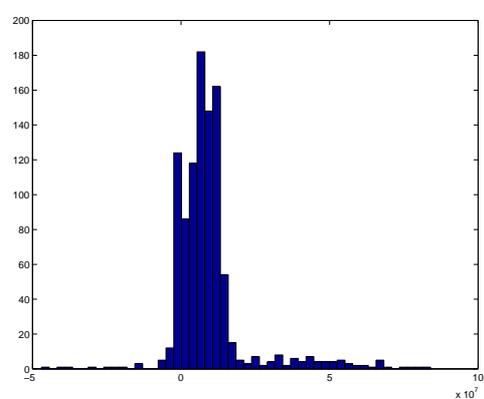
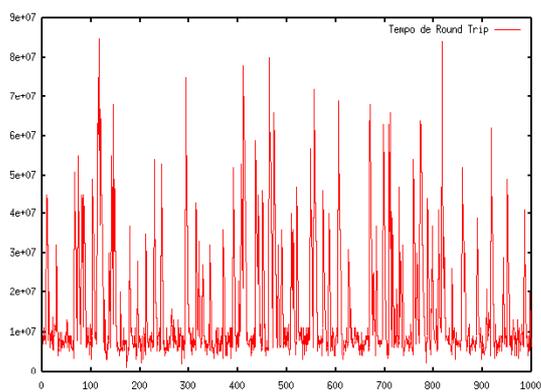
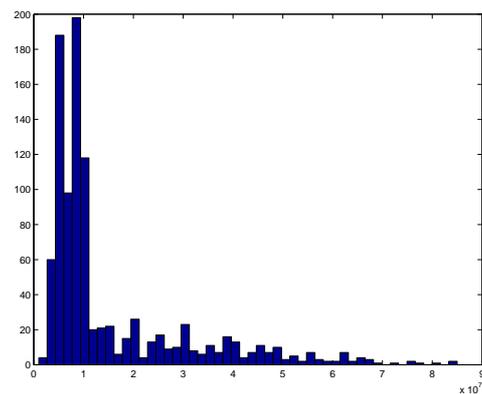
(c) *Jitter* do período (ns).(d) Histograma do *jitter* do período.

Figura 4.16: Período de recepção do sinal de controle com controle através da rede com atraso.

(a) Tempo de *round-trip* (ns).

(b) Histograma do período.

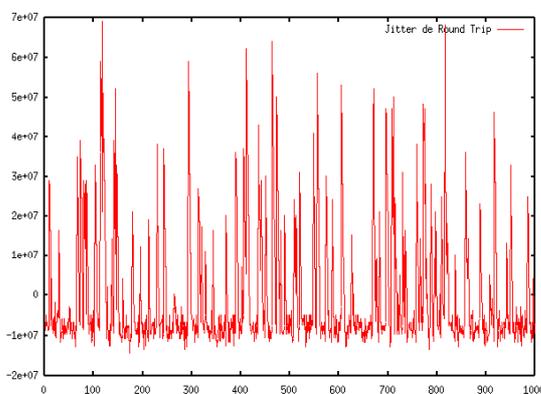
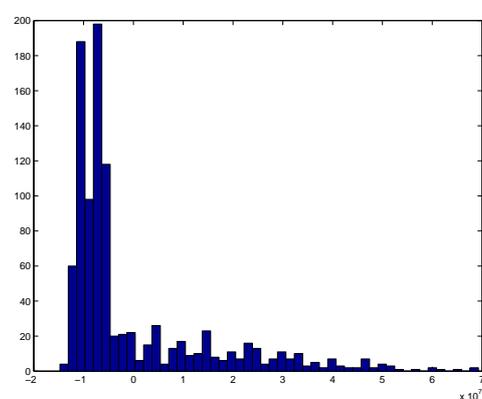
(c) *Jitter* do tempo de *round-trip* (ns).(d) Histograma do *jitter*.

Figura 4.17: Tempo de *round-trip* com controle através da rede com atraso.

(tabela 3.1), os indicadores medidos no sistema de controle com atrasos (tabela 4.8), indicam que as médias dos períodos de recepção, tanto do sinal de sensor como do sinal de controle, são similares ao sistema sem rede. No entanto, apresentam um desvio-padrão maior. Entretanto, o tempo de *round-trip* está acima do período de amostragem, comprometendo o desempenho do sistema, afetado ainda mais pelo desvio padrão que chega a ser maior do que 15 ms, causando instabilidade.

A figura 4.18 mostra as curvas de resposta do manipulador robótico com controlador em rede. A tabela 4.9 mostra algumas características de desempenho do sistema de controle em rede. Nota-se que não foi possível medir-se o tempo de acomodação tanto para a junta 1 como para a junta 2, apresentado no gráfico 4.18, devido ao mau desempenho do sistema de controle. A posição angular de referência para a junta 1 é 45° e para a junta 2 é 90° .

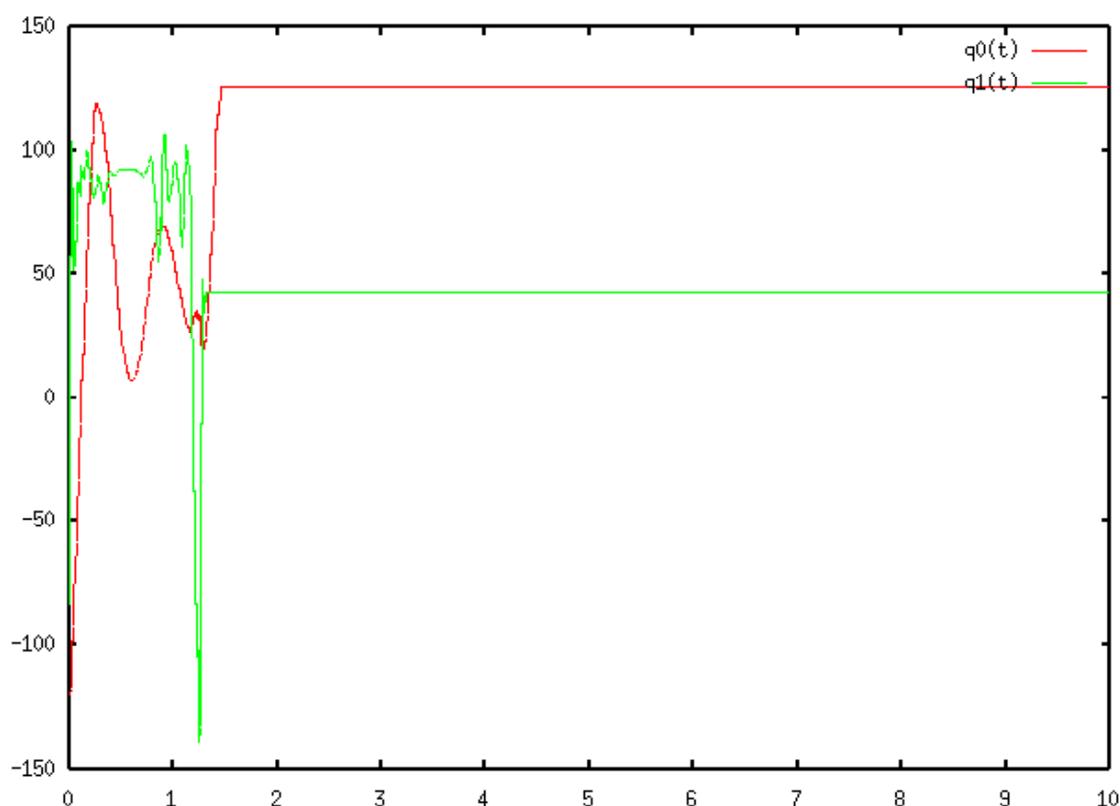


Figura 4.18: Resposta do controlador.

Pode-se verificar através da comparação das figuras 4.14 e 3.5, que os atrasos introduzidos pela rede IP, causam um impacto e degradação irreversíveis à operação do sistema de controle, resultando em uma resposta totalmente fora do desejado. Analisando os valores indicados através dos critérios de desempenho do sistema (ISE, IAE, ITSE e ITAE), indicados na tabela 4.9, em comparação aos valores obtidos na tabela 3.2, se verifica que o desempenho do controle foi seriamente prejudicado.

Tabela 4.9: Desempenho do sistema de controle através da rede com atraso.

	Junta 1	Junta 2
<i>Overshoot</i> (%)	44,93	6,46
Tempo de acomodação (segundos)	não medido	não medido
Tempo de subida (segundos)	0,16	0,02
Tempo de pico (segundos)	0,25	0,02
Critério ISE (radianos ²)	1802,181821	686,674517
Critério IAE (radianos)	1292,393529	760,289006
Critério ITSE (radianos ² × segundos)	9765,848334	3404,591958
Critério ITAE (radianos × segundos)	6943,237558	4066,603233

5 CONTROLE COM COMPENSAÇÃO DOS ATRASOS ROBÔ / CONTROLADOR

Neste capítulo, será apresentado o controle com compensação apenas do atraso no sentido robô/controlador. Este estudo tem por objetivo recuperar o desempenho do controlador local, impondo uma compensação nos atrasos sofridos pelos pacotes de rede.

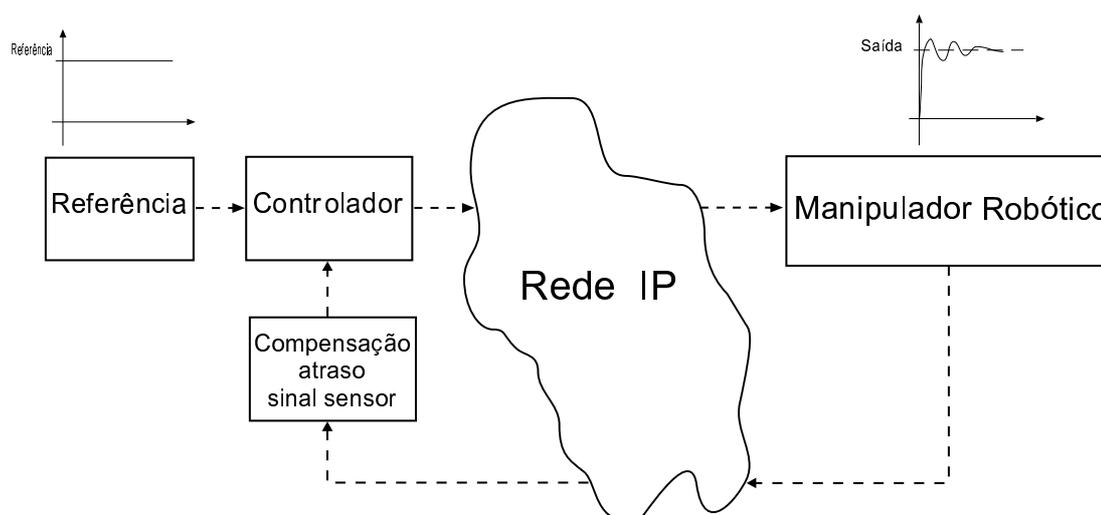


Figura 5.1: Estrutura do sistema de controle com compensação.

A compensação proposta, vide figura 5.1, é inserida entre o protocolo de transporte e a camada da aplicação responsável pelo sistema de controle. O mecanismo geral de controle utilizado na camada consiste na medida dos atrasos dos pacotes provenientes do sistema robótico compensando o atraso antes do cálculo da lei de controle. Isso é feito determinando-se o comportamento do robô em função de seu modelo (LAGES; ALT; HENRIQUES, 2003).

Obviamente, deseja-se compensar os atrasos tanto no sentido do manipulador robótico para o controlador, como também do controlador para o manipulador robótico. No entanto, as formas de compensação são aplicadas de forma distinta. O atraso dos pacotes do controlador para o manipulador robótico somente pode ser compensado no manipulador robótico, após o cálculo da lei de controle. As compensações, portanto, são realizadas de forma distintas. Neste capítulo 5, analisar-se-á

a compensação de atraso do sinal do sensor. A compensação dos atrasos controlador/robô será discutida no capítulo 6.

5.1 Medida dos atrasos

Para realizar as medidas dos atrasos do manipulador robótico e controlador, o pacote do protocolo de aplicação, apresentado na seção 4.4, inclui o instante de tempo de transmissão do pacote.

Com o tempo de transmissão, cada vez que o pacote chega ao seu destino, é realizado a medida do instante de tempo de recepção do pacote calculando o atraso do pacote através da diferença entre eles.

As técnicas de compensação são utilizadas quando o atraso é igual ou superior ao período de amostragem do sistema de 10 ms. Caso o pacote recebido contenha dados mais antigos que dados contidos em pacotes anteriormente recebidos, este pacote é descartado.

5.2 Atraso do sinal do sensor

Os valores lidos dos sensores do robô são transmitidos para o controlador no instante t_{ts} (vide figura 2.6), mas devido ao atraso da rede, os mesmos chegam ao controlador no instante $t_{rs} > t_{ts}$. A transmissão do sinal do sensor é realizada de forma periódica.

A compensação do atraso proposta consiste em se obter uma estimativa dos valores dos sensores que seriam lidos no instante de cálculo da lei de controle, ou seja, no instante t_{rs} . Isto pode ser feito propagando-se o valor recebido, que foi lido e transmitido no instante t_{ts} , através do modelo dinâmico do robô (3.14). Ou seja, tem-se

$$\hat{x}(t_{rs}) = x(t_{ts}) + \int_{t_{ts}}^{t_{rs}} f(x) + g(x)u dt \quad (5.1)$$

onde:

$\hat{x}(t_{rs}) =$ Estimativa de x no instante t_{rs} ;

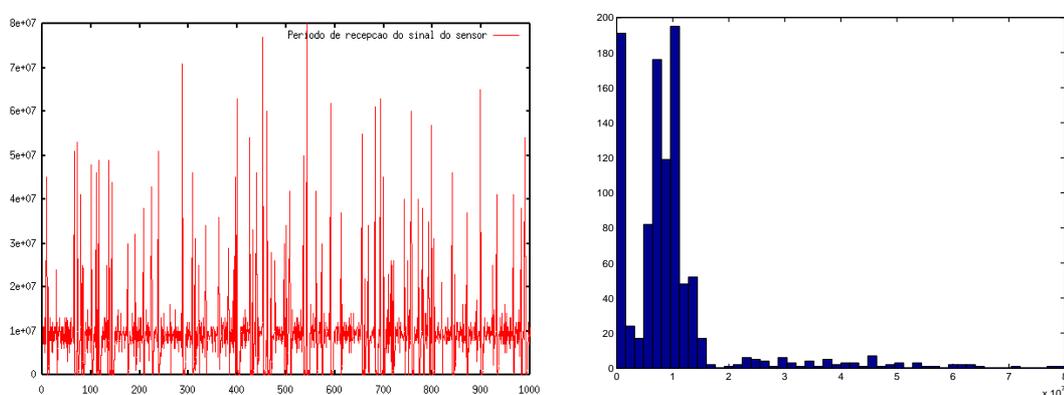
$x(t_{ts}) =$ Valor de x no instante t_{ts} (valores lidos dos sensores);

A integral da expressão (5.1) pode ser computada numericamente, por exemplo utilizando-se o método de Runge-Kutta (PRESS et al., 1992). É importante notar que integrais numéricas são procedimentos iterativos e, portanto, não muito aconselháveis em tempo real. Neste caso, serão executados apenas alguns passos de integração para propagar x do instante t_{ts} até o instante t_{rs} .

O valor de $\hat{x}(t_{rs})$ é utilizado para cálculo do sinal de controle através da expressão (3.15) como se fosse o valor de x lido no instante t_{rs} . Note-se que, desprezando-se o tempo de cálculo da lei de controle, tem-se $t_{rs} = t_{tc}$.

5.3 Resultados

Com o controle e o manipulador robótico operando em rede com atraso, o período de recepção do sinal do sensor sofre variações devido aos atrasos da comunicação no sentido manipulador robótico para controlador. A figura 5.2 mostra os gráficos e os histogramas do período e do *jitter* de recepção do sinal de controle. Pode-se notar que, mesmo com maior média do tempo de recepção de pacotes com o sinal de controle, em comparação a figura 4.15, a resposta do sistema de controle, através da rede com compensação de atraso do sinal do sensor, é mais aproximado à resposta do sistema de controle através da rede sem atraso. Destaca-se também que o período está próximo do limite de operação do sistema que, mesmo assim, apresenta uma significativa melhora de desempenho do sistema (vide tabela 5.1), em relação ao sistema de controle com atraso sem compensação (vide tabela 4.8).



(a) Período (ns).

(b) Histograma do período.

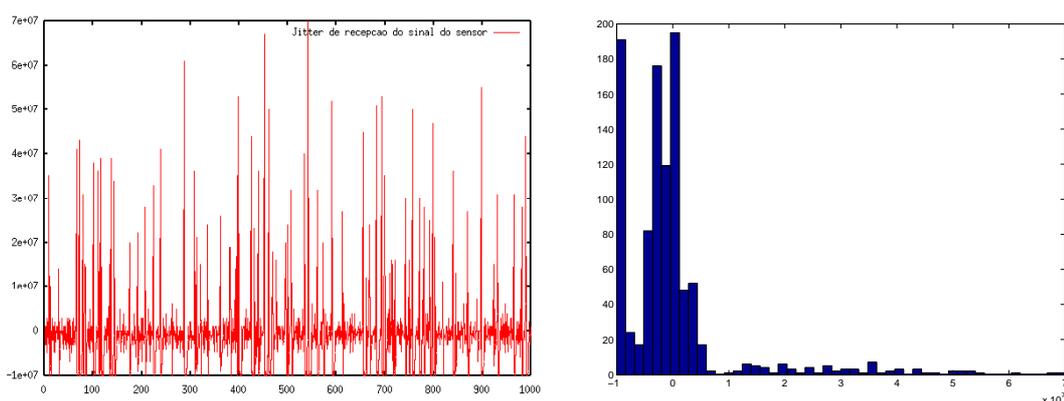
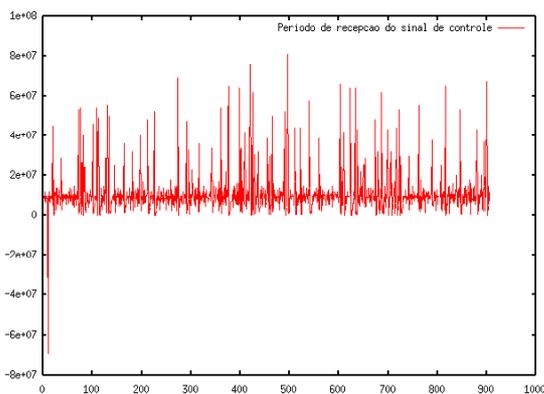
(c) *Jitter* do período (ns).(d) Histograma do *jitter* do período.

Figura 5.2: Período de recepção do sinal de controle com compensação de atraso do sinal do sensor.

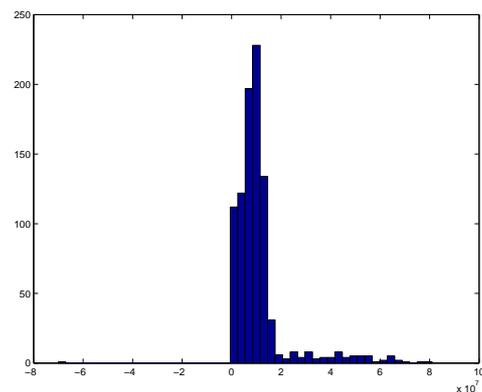
O valor de referência para a junta 1 é 45° e para a junta 2 é 90° .

O período de recepção do sinal de controle sofre variações devido aos atrasos da comunicação no sentido manipulador robótico para controlador e no sentido contro-

lador para manipulador robótico. A figura 5.3 mostra os gráficos e os histogramas do período e do *jitter* de recepção do sinal de controle. Importante notar que, em comparação com o experimento do controle através da rede com atraso sem compensação (vide figura 4.16), a resposta do sistema de controle através da rede apresentou um melhor desempenho. Com a compensação aplicada no controlador, há um descarte de 9,4 % dos pacotes contendo o sinal de controle que, somado ao atraso da rede, acarretam uma tendência de degradação de desempenho do controle, que é amenizada pela compensação de atraso do sinal do sensor no controlador.



(a) Período (ns).



(b) Histograma do período.

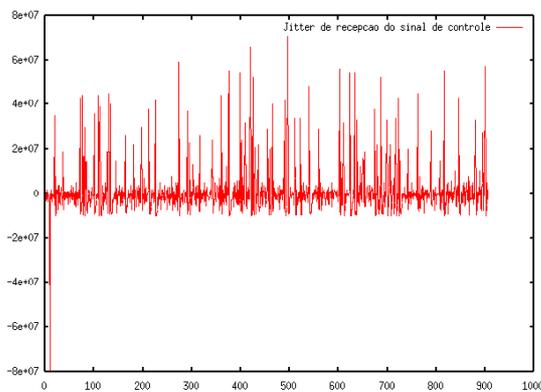
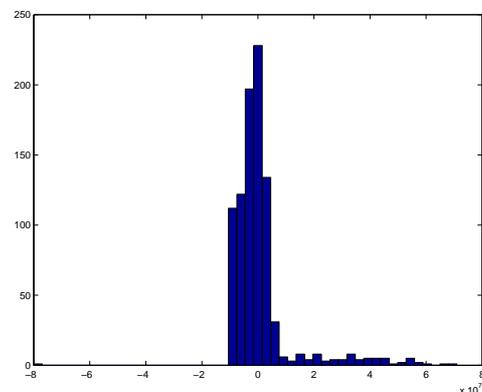
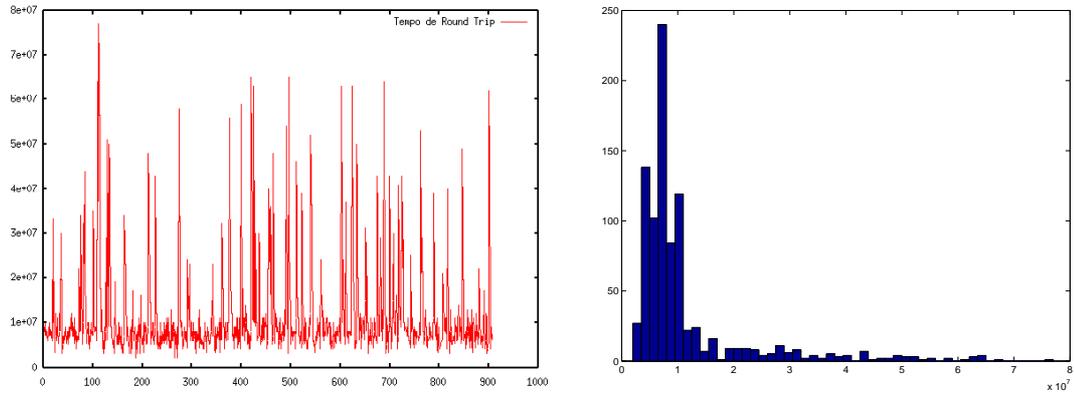
(c) *Jitter* do período (ns).(d) Histograma do *jitter* do período.

Figura 5.3: Período de recepção do sinal de controle com compensação de atraso do sinal do sensor.

A figura 5.4 mostra os tempos de *round-trip* dos sinais de controle e sensor através da rede com atraso. A tabela 5.1 mostra algumas características de desempenho do sistema de controle em rede. O sistema, mesmo operando com atrasos acima do período de amostragem do sistema, apresentou um desempenho mais próximo da resposta de controle ideal.

A figura 5.5 mostra as curvas de resposta com controlador em rede com compensação dos atrasos do robô/controlador para uma situação de carga de rede se-

(a) Tempo de *round-trip* (ns).

(b) Histograma do período.

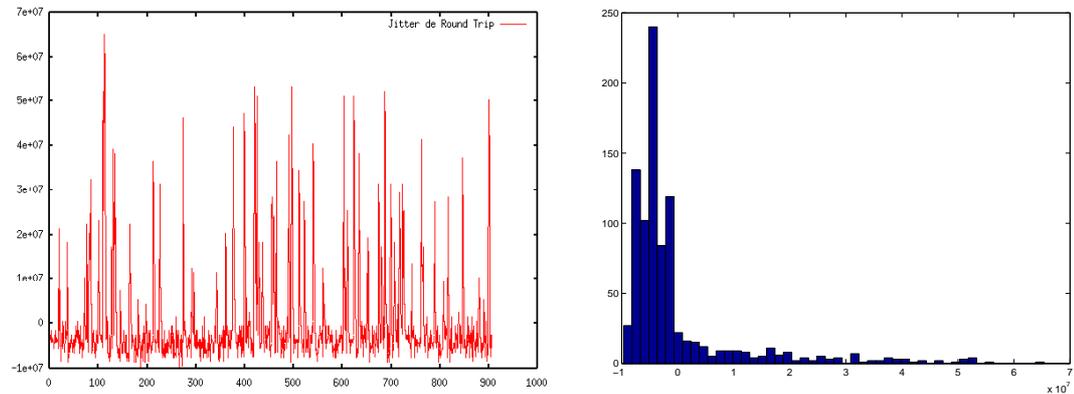
(c) *Jitter* do tempo de *round-trip* (ns).(d) Histograma do *jitter*.Figura 5.4: Tempo de *round-trip* com compensação do atraso do sinal do sensor.

Tabela 5.1: Estatísticas das métricas com controle através da rede com compensação de atraso do sinal do sensor.

Métrica	Média (segundos)	Desvio Padrão (segundos)
T_{rs}	$9723729,72 \times 10^{-09}$	$10412885,09 \times 10^{-09}$
J_{rs}	$-276276,28 \times 10^{-09}$	$10412885,09 \times 10^{-09}$
T_{rc}	$10719337,02 \times 10^{-09}$	$11586366,14 \times 10^{-09}$
J_{rc}	$719337,02 \times 10^{-09}$	$11586366,14 \times 10^{-09}$
T_{rt}	$11654525,39 \times 10^{-09}$	$11132696,83 \times 10^{-09}$
J_{rt}	0	$11132696,83 \times 10^{-09}$

melhante e até mais desfavorável com a qual foi obtida a resposta do controlador, no sistema de controle com atraso sem compensação (vide figura 4.18). A posição angular de referência para a junta 1 é 45^0 e para a junta 2 é 90^0 . A tabela 5.2 mostra alguns indicadores de desempenho do sistema de controle obtidos com a compensação dos atrasos do robô/controlador.

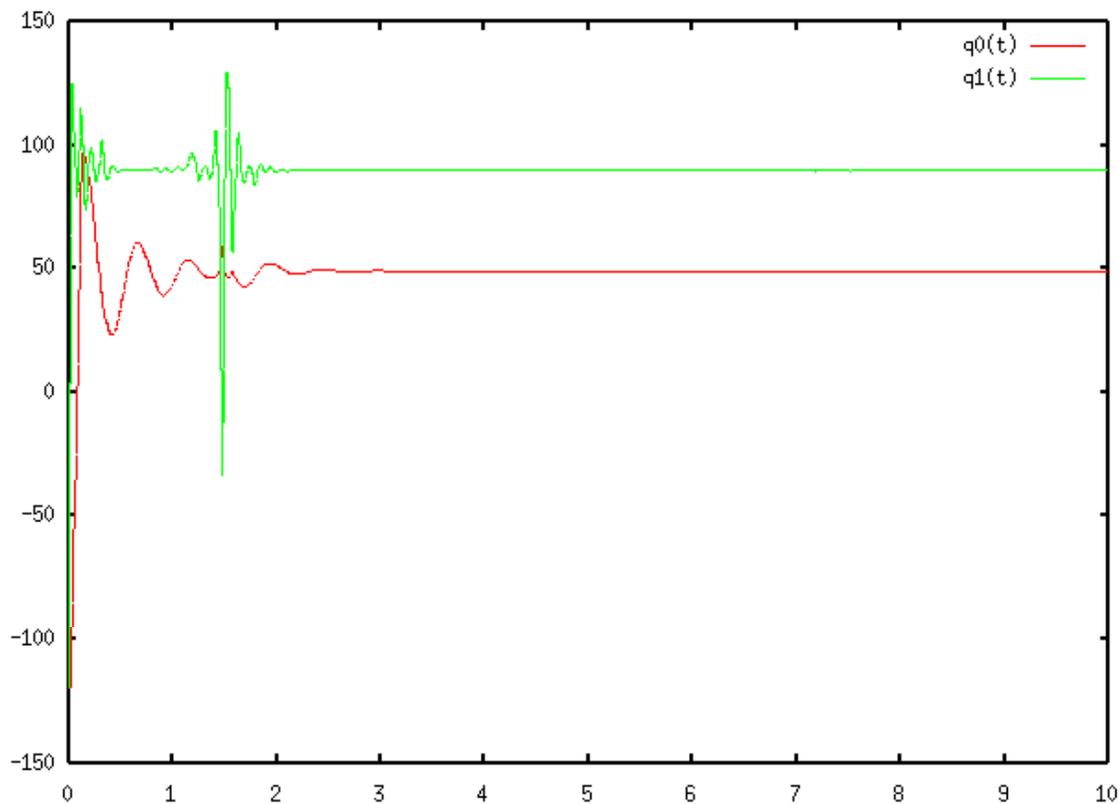


Figura 5.5: Resposta do controlador com compensação dos atrasos robô / controlador.

Em comparação com os indicadores apresentados no experimento realizado com controle através da rede sem atraso (vide tabela 4.7), há uma sensível aproximação do desempenho devido à compensação de atraso do sinal do sensor (vide tabela 5.2). Verifica-se, principalmente, um melhor desempenho da junta 1, que possui os mecanismos mais pesados que a junta 2, apresentando, portanto, uma inércia maior e, assim, uma maior tolerância às variações de atrasos e perda de pacotes.

Como esperado, a compensação dos atrasos do sinal do sensor melhora o desempenho do sistema de controle. No entanto, apenas os atrasos em um dos sentidos da comunicação foram compensados, sendo portanto razoável esperar um desempenho ainda melhor ao compensar também os atrasos no sentido controlador/robô.

Tabela 5.2: Desempenho do sistema de controle com compensação dos atrasos robô/controlador.

	Junta 1	Junta 2
<i>Overshoot</i> (%)	31,23	16,74
Tempo de acomodação (segundos)	2,15	1,82
Tempo de subida (segundos)	0,108	0,025
Tempo de pico (segundos)	0,13	0,03
Critério ISE (radianos ²)	60,699671	35,932496
Critério IAE (radianos)	105,430278	29,450118
Critério ITSE (radianos ² × segundos)	28,675426	21,110612
Critério ITAE (radianos × segundos)	356,038187	27,937111

6 CONTROLE COM COMPENSAÇÃO DOS ATRASOS ROBÔ / CONTROLADOR E CONTROLADOR / ROBÔ

Neste capítulo, será apresentado o sistema de controle com compensação dos atrasos do manipulador robótico / controlador e controlador / manipulador robótico. Esta combinação de técnicas tem por objetivo recuperar o desempenho obtido com o sistema de controle local, na tentativa de reduzir os efeitos dos atrasos dos pacotes que trafegam na rede, impondo uma compensação dos atrasos, tanto do sinal de controle como do sinal do sensor. Com os resultados obtidos, analisa-se o comportamento desta estratégia em relação ao controle através da rede sem qualquer compensação.

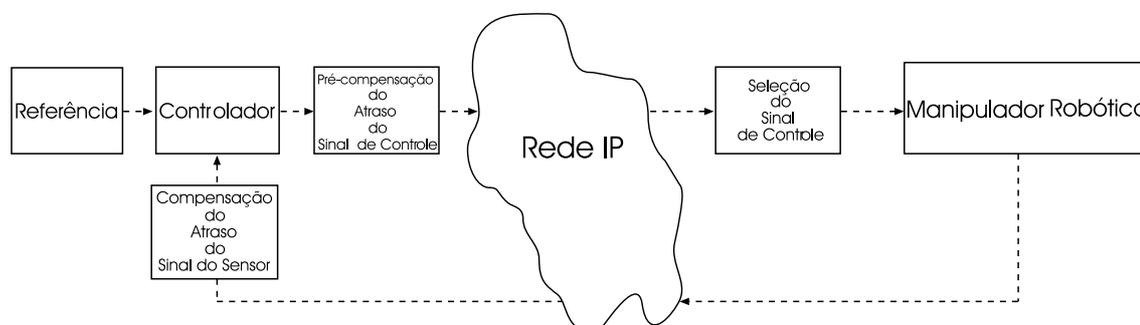


Figura 6.1: Controle com compensação dos atrasos robô/controlador e do atraso controlador/robô.

6.1 Atraso do sinal de controle

Diferentemente do atraso do robô/controlador, o atraso do controlador/robô não é conhecido no instante de cálculo da lei de controle, o que impede que se obtenha um valor do controle a ser aplicado, já incluindo a compensação. Obviamente, o atraso controlador/robô pôde ser determinado pelo nodo do manipulador robótico utilizando o *time-stamp* dos pacotes. Porém, não há muito sentido fazer o nodo do manipulador robótico executar os complexos cálculos que seriam necessários para computar uma estimativa de u , com o atraso compensado através de um método semelhante ao utilizado para compensar o atraso do robô/controlador, pois seria

algo equivalente a calcular a lei de controle localmente. Assim, para que o controle através da rede seja justificável e possa explorar as suas vantagens, é necessário que a complexidade adicional no nodo do manipulador robótico seja muito pequena. Este objetivo pode ser atingido fazendo com que o controlador pré-compense o atraso do sinal do sensor.

Como o atraso do sinal de controle é desconhecido pelo nodo controlador, pode-se computar o sinal de controle compensado para diversos valores de atraso e transmiti-los para o manipulador robótico. O nodo do manipulador robótico, após determinar o atraso do sinal de controle, seleciona o valor do sinal de controle com a compensação adequada e aplica-o ao robô. Ou seja, o controle u transmitido pelo controlador passa a ser uma matriz $n \times m$, onde n é o número de juntas do robô e m é o número de passos a frente para os quais será calculada compensação de atraso do sinal de controle, conforme a expressão (6.1).

$$u = \begin{bmatrix} u_0 & u_1 & \cdots & u_m \end{bmatrix} \quad (6.1)$$

onde cada u_k é determinado pela expressão (3.16) propagada k passos a frente, ou seja,

$$u_k = h(\hat{x}(t_{tc} + kT_s), q_r(t_{tc} + kT_s), \dot{q}_r(t_{tc} + kT_s), \ddot{q}_r(t_{tc} + kT_s)) \quad (6.2)$$

sendo T_s o período de amostragem do controle e as estimativas $\hat{x}(t_{tc} + kT_s)$ dadas pela propagação k passos a frente das leituras dos sensores já compensadas com relação ao atraso do sinal do sensor. Ou seja,

$$\hat{x}(t_{tc} + (k+1)T_s) = \hat{x}(t_{tc} + kT_s) + \int_{t_{tc} + kT_s}^{t_{tc} + (k+1)T_s} f(x) + g(x)u_k dt \quad (6.3)$$

com $\hat{x}(t_{tc}) = \hat{x}(t_{rs})$. Isto é, a expressão (6.3) seria a propagação para o futuro da compensação de atraso do sinal de controle/sensor realizada através da expressão (5.1). No entanto, ao ser utilizada conjuntamente com a expressão (6.3), a expressão (5.1) necessita considerar que o sinal de controle efetivamente aplicado ao manipulador robótico não é mais o computado através da expressão (3.16) para o instante t_{tc} , como na seção 5.2. Agora, o sinal de controle efetivamente aplicado é escolhido pelo nodo do manipulador robótico entre os diversos valores de u_k . Ou seja, a expressão (5.1) é modificada para:

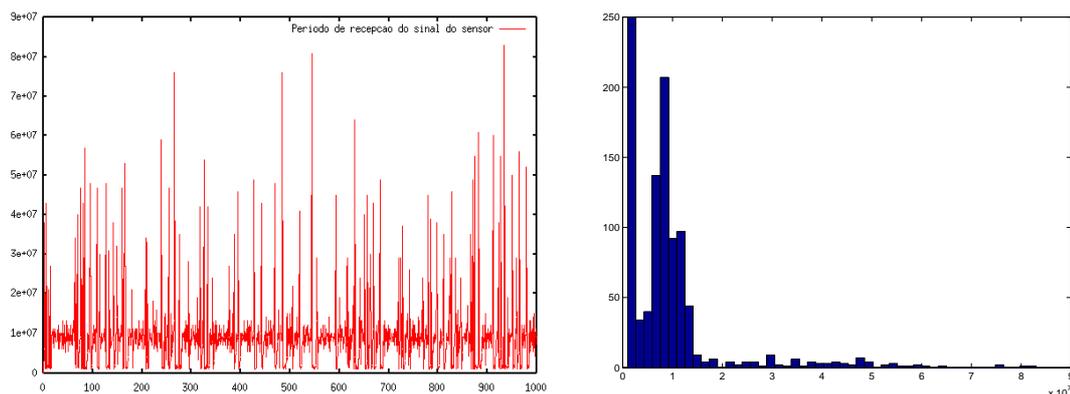
$$\hat{x}(t_{rs}) = x(t_{ts}) + \int_{t_{ts}}^{t_{rs}} f(x) + g(x)u_{k*} dt \quad (6.4)$$

onde $u_{k*} = u(t_{ts})$ é o sinal de controle aplicado no intervalo $(t_{ts}, t_{ts} + T_s)$.

6.2 Resultados

Os experimentos foram realizados utilizando o manipulador robótico e controlador, realizando a comunicação através da rede, com compensação de atraso do sinal do sensor e com compensação de atraso do sinal de controle, ou seja, tanto no sentido robô/controlador como vice-versa.

Com o controle e o manipulador robótico operando em rede com atraso, este período sofre variações devido aos atrasos da comunicação no sentido robô para controlador. A figura 6.2 mostra os gráficos e os histogramas do período e do *jitter* de recepção do sinal do sensor. Verifica-se que a média do tempo de recepção de pacotes com o sinal de controle é similar aos experimentos do sistema de controle com a rede com atraso (figura 4.15).



(a) Período (ns).

(b) Histograma do período.

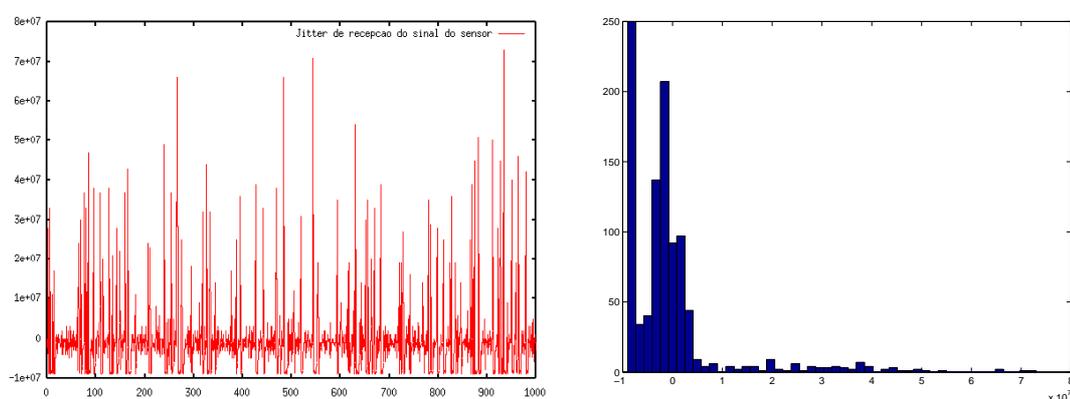
(c) *Jitter* do período (ns).(d) Histograma do *jitter* do período.

Figura 6.2: Período de recepção do sinal de sensor com compensações dos atrasos robô / controlador e controlador / robô.

O valor de referência para a junta 1 é 45° e para a junta 2 é 90° .

A figura 6.3 mostra os gráficos e os histogramas do período e do *jitter* de recepção do sinal de controle. Importante notar que, em comparação com a figura 5.3, mesmo apresentando uma média maior, a resposta do sistema de controle através da rede apresentou um melhor desempenho. Com a compensação aplicada no controlador e no manipulador robótico, há um descarte de 9 % dos pacotes contendo o sinal de controle que, somado ao atraso da rede, acarreta uma tendência de degradação de desempenho do manipulador robótico, mas que é compensado pelo envio antecipado do sinal de controle a ser aplicado nas juntas, reduzindo significativamente o impacto

dos pacotes perdidos durante a transmissão do sinal de controle, minimizando o efeito negativo sobre o desempenho de operação do manipulador robótico.

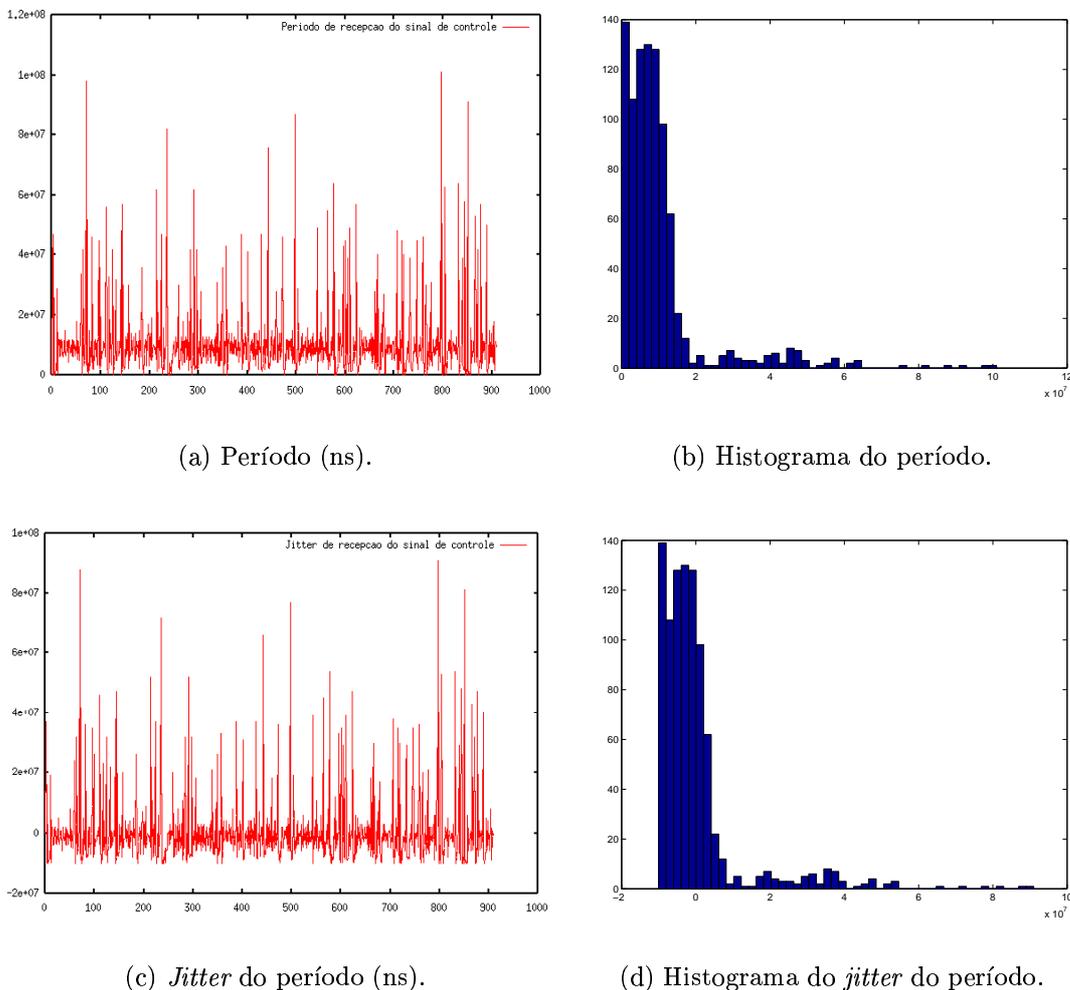


Figura 6.3: Período de recepção do sinal de controle com compensações dos atrasos robô/controlador e controlador/robô.

O tempo de *round-trip* também sofre influência dos atrasos introduzidos, tanto no sentido robô para o controlador, como no sentido controlador para o robô. No entanto, o tempo de *round-trip* de cada período de amostragem não sofre influência dos atrasos sofridos nos instantes anteriores, como acontece com o período de recepção do sinal de controle. A figura 4.17 mostra os gráficos e os histogramas do tempo e do *jitter* de *round-trip*.

A tabela 5.1 mostra algumas características de desempenho do sistema de controle em rede que, em comparação ao experimento do controle sem compensação, vide tabela 4.8, mostra condições desfavoráveis para a operação do sistema neste experimento. A média do tempo de *round-trip* apresentou uma média superior; mas, mesmo nestas circunstâncias, a resposta do sistema de controle, vide figura 6.5, foi mais próxima da resposta do controle local, vide figura 3.5, apesar de ainda se verificar o impacto de atrasos na rede na resposta do controle.

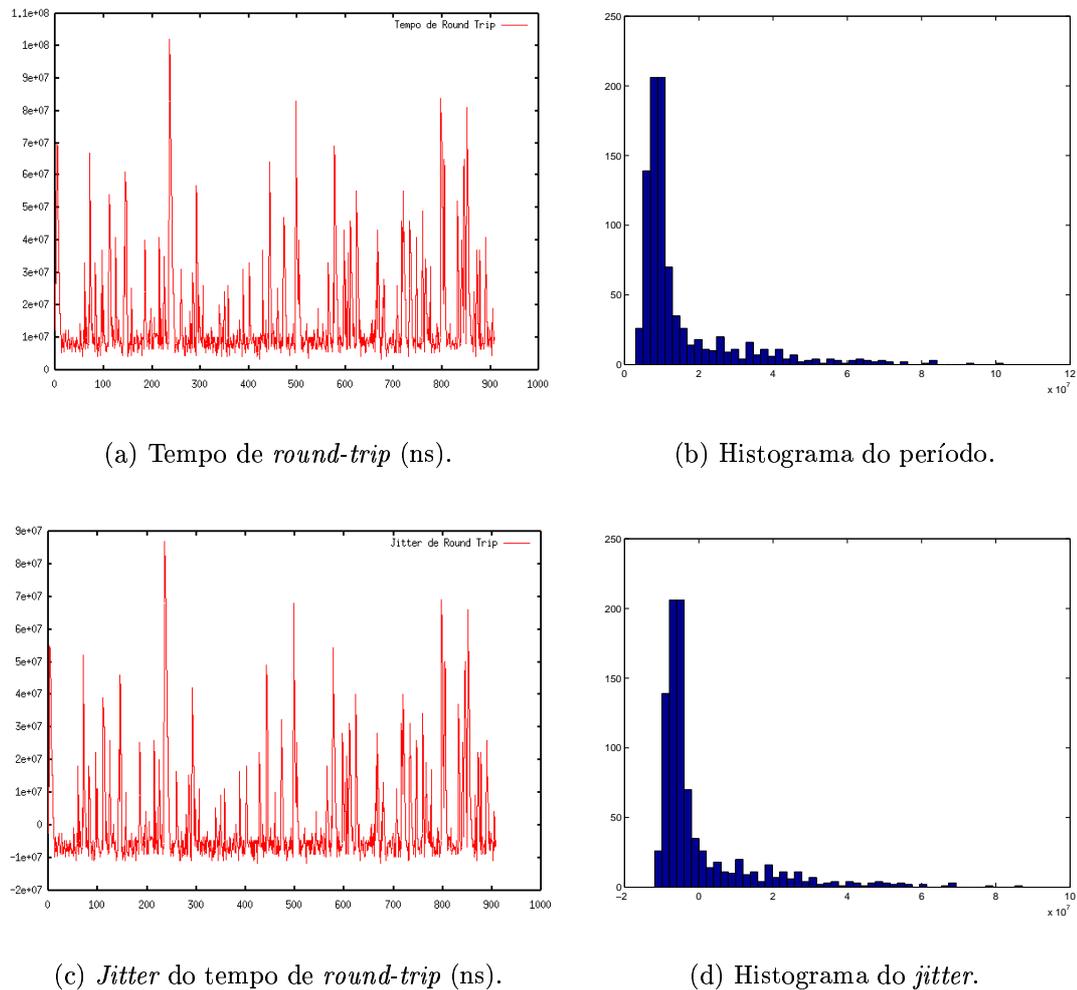


Figura 6.4: Tempo de *round-trip* com compensações dos atrasos robô / controlador e controlador / robô.

Tabela 6.1: Estatísticas das métricas computadas para o controle com rede carregada com compensações no controlador e no robô.

Métrica	Média (segundos)	Desvio Padrão (segundos)
T_{rs}	$9477477,48 \times 10^{-09}$	$10763786,62 \times 10^{-09}$
J_{rs}	$-522522,52 \times 10^{-09}$	$10736050,26 \times 10^{-09}$
T_{rc}	$10414741,47 \times 10^{-09}$	$12332068,55 \times 10^{-09}$
J_{rc}	$414741,47 \times 10^{-09}$	$12332068,55 \times 10^{-09}$
T_{rt}	$14868131,87 \times 10^{-09}$	$14358065,31 \times 10^{-09}$
J_{rt}	0	$14358065,31 \times 10^{-09}$

A figura 6.5 mostra as curvas de resposta com controlador em rede com compensação dos atrasos robô/controlador e controlador/robô com uma simulação da carga de rede semelhante e até mais desfavorável com a qual foi obtida a figura 4.18. A posição angular de referência para a junta 1 é 45° e para a junta 2 é 90° . A tabela 6.2 mostra algumas características de desempenho do sistema de controle com as compensações aplicadas.

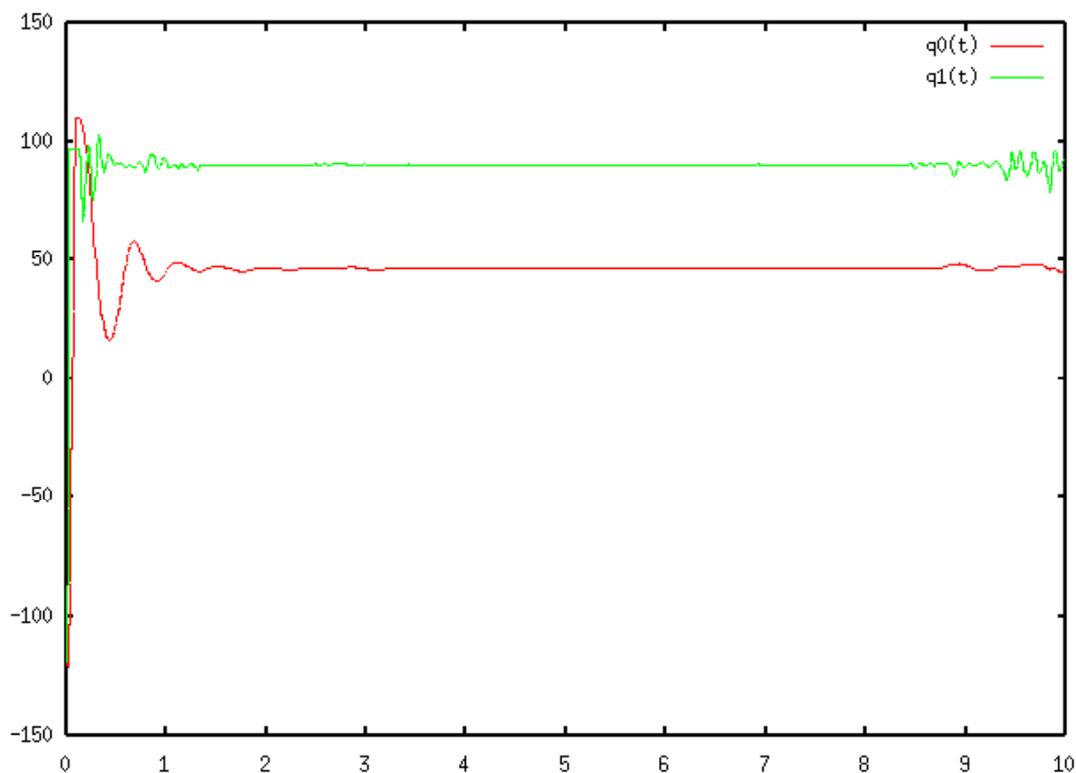


Figura 6.5: Resposta do controlador com compensações no controlador e no robô.

A figura 6.5 mostra a resposta do manipulador com compensações dos atrasos nos dois sentidos para uma situação de carga de rede semelhante à que foi utilizada na obtenção das figuras 5.5 e 4.18. A tabela 6.2 mostra algumas características de desempenho do sistema de controle com compensação dos atrasos. Novamente, como antecipado, tem-se uma melhora de desempenho do sistema de controle em relação aos experimentos onde não se contemplou compensação dos atrasos, seção 4.6.1, ou mesmo se aplicando a compensação em apenas um sentido, seção 5.3.

Em comparação aos indicadores de desempenho dos testes do controle através da rede com as compensações de atraso do sinal de controle e sensor, indicado na figura 6.5, há uma maior aproximação ao desempenho do sistema com controle sem rede (figura 3.5). Verificam-se também significativas melhoras em relação aos experimentos do modelo de compensação de um sentido, atraso do sinal do sensor. Analisando a tabela 6.2, nota-se um melhor desempenho dos indicadores, comprovando um desempenho mais próximo ao obtido no sistema de controle local, apresentado na tabela 3.2. A variação observada no final do gráfico apresentado na figura 6.5 significa a incidência de atrasos na rede no final da amostragem dos pacotes dos sinais

Tabela 6.2: Desempenho do sistema de controle com compensações dos atrasos robô/controlador e controlador/robô.

	Junta 1	Junta 2
<i>Overshoot</i> (%)	39,40	3,27
Tempo de acomodação (segundos)	1,58	1,33
Tempo de subida (segundos)	0,074	0,03
Tempo de pico (segundos)	0,09	0,03
Critério ISE (radianos ²)	64,170254	34,656327
Critério IAE (radianos)	70,404001	26,216694
Critério ITSE (radianos ² × segundos)	9,271035	4,071319
Critério ITAE (radianos × segundos)	137,800391	70,026397

de controle/sensor, cujos efeitos foram minimizados pela aplicação das técnicas de compensações. Percebe-se que os valores dos critérios, em comparação ao indicado no experimento com controle local (tabela 4.7), apresentam um melhor desempenho que o obtido com o controlador local, especialmente para a junta 1. Entretanto, a junta 2, por ser mais leve que a junta 1, apresenta uma inércia menor e, assim uma menor tolerância a variações de atrasos e perda de pacotes, ficando muito mais susceptível aos atrasos do sinal de controle. Todavia, os indicadores obtidos neste experimento se aproximam muito e até superam, em alguns indicadores apresentados, o desempenho do sistema de controle local. Com isso, se permite uma nova perspectiva de qualidade ao sistema mesmo com os atrasos verificados, satisfazendo as condições para a operação do controle através da rede IP.

7 CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS

Redes de controle diferem das redes de dados em muitos pontos, como na transmissão de pequenos pacotes com requisitos de tempo real, pacotes de pequeno tamanho de dados e mensagens periódicas, garantia de transmissão e limite no tempo de atraso. Sinais são enviados através da rede e reconstruídos na outra extremidade, onde são utilizados para atuar no controle do sistema. A rede de controle afeta o desempenho da malha fechada devido às diferenças entre os sinais enviados e os sinais que chegam para controle do sistema, pelo atraso imposto pela rede.

Com os distintos cenários apresentados neste trabalho, entre o sistema sem rede e a separação do controle e o manipulador robótico com uma rede IP entre os dois, se verifica que a ocorrência de atrasos não determinísticos do controlador para o manipulador robótico e vice-versa, provocam distúrbios na resposta do controle. Os atrasos foram caracterizados e foram propostas técnicas de compensações de atrasos de rede, que se mostraram capazes de recuperar boa parte do desempenho perdido com relação a um sistema de controle local.

O desempenho verificado nos experimentos tabela 7.1 indica que o segundo método de compensação proposto, compensações dos atrasos robô / controlador e controlador / robô, em comparação com o método de compensação apenas de atrasos do robô / controlador, apresenta um melhor desempenho.

Os valores apresentados, indicados através dos critérios de desempenho do sistema de controle na tabela 7.1, permitem avaliar o desempenho do sistema nas mesmas condições. No caso ideal, tem-se o desempenho do controle local. A partir da introdução da rede na malha de controle, percebe-se um aumento nos índices indicando uma piora no desempenho, embora verifique-se que o comportamento do controle através da rede sem atraso é muito similar ao verificado no sistema com controle local. Este indica que, na ausência de atrasos excessivos, o controle via rede é viável. No caso do controle através da rede com atraso verifica-se, através da comparação, que há degradação no desempenho do sistema de controle. Sem técnica de compensação dos atrasos aplicada no sistema, o controle não responde adequadamente. A partir da aplicação de compensação, em um sentido apenas, o sistema responde mais eficientemente e se aproxima dos índices obtidos com controle local. Por fim, com o controlador com compensações tanto no sentido robô para o controlador, como vice-versa, obtém-se uma maior aproximação ao desempenho do sistema com controle local, satisfazendo as condições para a operação do controle

Tabela 7.1: Desempenho dos sistemas de controle através da rede.

	Junta	ISE	IAE	ITSE	ITAE
Controle local	1	79,7715	69,8436	10,9993	57,0757
	2	11,4425	11,6929	0,1628	23,8816
Controle através da rede sem atraso	1	79,7811	70,3453	11,0653	61,0415
	2	11,4425	11,6908	0,1628	23,8576
Controle através da rede com atraso	1	1802,18	1292,39	9765,84	6943,23
	2	686,67	760,28	3404,59	4066,60
Controle com compensação robô / controlador	1	60,6996	105,4302	28,6754	356,0381
	2	35,9324	29,4501	21,1106	27,9371
Controle com compensações robô / controlador e controlador / robô	1	64,1702	70,4040	9,2710	137,8003
	2	34,6563	26,2166	4,0713	70,0263

através da rede IP.

Para redes com atrasos superiores ao período de amostragem, a pré-compensação do atraso do sinal de controle se mostrou capaz de tratar os problemas de sincronização entre o controlador e o manipulador robótico, assegurando a correta operação do sistema.

Para o critério ITAE, a figura 7.1 apresenta um gráfico de barras com uma comparação das estratégias de controle através da soma dos valores do critério para as duas juntas. Para os demais critérios, a visualização através de gráfico de barras torna-se pouco ilustrativa, devido à falta de seletividade destes critérios.

Este trabalho serve como base para se implementar serviços mais eficientes e que garantam uma maior qualidade de resposta e, assim, possam viabilizar técnicas que permitam implementar o conceito de telepresença para o controle do sistema robótico através da Internet, possibilitando a interação em tempo real. Dois métodos para comparar os atrasos impostos pela rede foram apresentados: Controle com compensação dos atrasos no sentido robô / controlador (capítulo 5) e Controle com compensação dos atrasos nos sentidos do robô / controlador e controlador / robô (capítulo 6), mantendo o desempenho, mesmo com os atrasos decorrentes dos atrasos da comunicação entre o controlador e o robô.

A utilização de redes será cada vez mais comum. O sistema de controle será familiar e acessível a quem esteja conectado a uma rede IP. A área de sistemas de controle distribuído tem muito a oferecer. Com uma taxa de rede suficientemente compatível com a necessidade do sistema distribuído, somado às técnicas apresentadas, será suficiente para suportar a operação de sistemas em tempo real. A pesquisa nesta área promete uma combinação interessante entre as redes de computadores e a teoria de controle. A área de abrangência da utilização destas técnicas traz inúmeros benefícios às áreas de automação industrial, automação de edifícios, escritórios e casas, bem como sistemas de veículos inteligentes e projetos aeroespaciais.

Como seqüência deste trabalho, existe a possibilidade de estudar a utilização de técnicas mais sofisticadas para estimação da compensações a serem aplicadas nos sinais dos sensores e de controle; em especial, a utilização do filtro de Kalman (KAL-

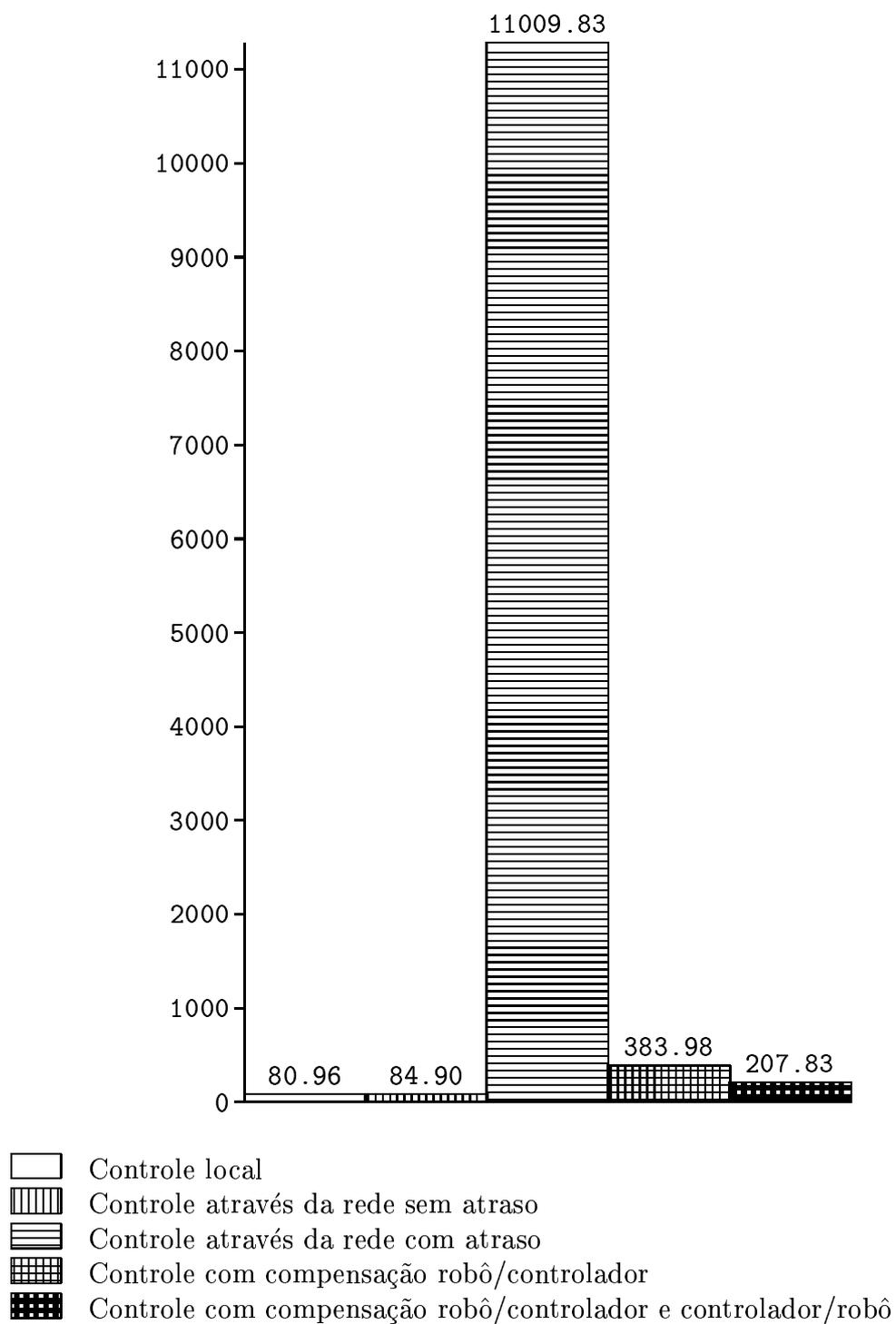


Figura 7.1: Comparação das estratégias de controle pelo critério ITAE.

MAN, 1960) para obter-se estimativas ótimas que aprimorem os resultados obtidos neste trabalho. Sugere-se também a utilização de técnicas de QoS com objetivo de garantir um limite máximo para os atrasos. Como os problemas são causados não só pelos atrasos em si, mas principalmente por serem variáveis, as técnicas propostas aqui cumprem uma finalidade diferente das técnicas de QoS. Outra sugestão seria a utilização da versão do protocolo IP versão 6, uma vez que neste trabalho se desenvolveu uma estrutura que permite utilizar outros protocolos, sem ter que alterar o controlador ou mesmo a interface com o robô. Outros trabalhos consistiriam na utilização de uma arquitetura totalmente distribuída, com um processador para cada junta, tendo em vista que o protocolo desenvolvido já está preparado para receber esta implementação.

Um das contribuições mais significativa deste trabalho, é explorar a arquitetura aberta para *retrofitting* de robôs (ALT; LAGES, 2003c), visando disponibilizar a robotização mais acessível para empresas de pequeno e médio porte, que hoje estão à margem de adquirir as tecnologias proprietárias, de alto custo. Esta redução de custo é permitida em função da utilização de sistemas de arquitetura aberta para o controle do robô, com a utilização de robôs usados, atualizando apenas o seu *software*, uma vez que o *hardware* é aproveitado.

REFERÊNCIAS

- ALT, G. H.; LAGES, W. F. An Assessment of Real-Time Robot Control over IP Networks. In: REAL-TIME LINUX WORKSHOP, 4., 2002, Boston, USA. **Proceedings...** [S.l.]: Real-Time Linux Foundation, 2002. p.1–17.
- ALT, G. H.; LAGES, W. F. Avaliação de Sistema de Controle em Tempo Real Através da Rede IP. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 6., 2003, Bauru, SP, Brasil. **Anais...** [S.l.]: SBA, 2003. p.355–360.
- ALT, G. H.; LAGES, W. F. Networked Robot Control With Delay Compensation. In: REAL-TIME LINUX WORKSHOP, 5., 2003, Valencia, Spain. **Proceedings...** [S.l.]: Real-Time Linux Foundation, 2003.
- ALT, G. H.; LAGES, W. F. Controle em Tempo Real de Sistemas Dinâmicos Através de Redes IP. In: CONGRESSO INTERNACIONAL DE AUTOMAÇÃO, SISTEMAS E INSTRUMENTAÇÃO, 3., 2003, São Paulo, SP, Brasil. **Anais...** [S.l.]: ISA, 2003.
- ASADA, H.; SLOTINE, J.-J. E. **Robot Analysis and Control**. New York, USA: Jonh Wiley & Sons, 1986.
- BEJCZY, A. K. Teleoperators. **IEEE Industrial Electronics Society Newsletters**, Piscataway, NJ, USA, p.4–12, 1996.
- BIANCHI, E.; DOZIO, L. Some Experiences in fast hard real-time control in user space with RTAI-LXRT. In: REAL-TIME LINUX WORKSHOP, 2000, Orlando, USA. **Proceedings...** [S.l.]: Real-Time Linux Foundation, 2000.
- BROCKETT, R. Stabilization of motor networks. In: IEEE CONFERENCE DECISION AND CONTROL, 1995. **Proceedings...** [S.l.]: IEEE Control Systems Society, 1995. p.1484–1488.
- BUSHNELL, L. Special Section - Networks and Control. **IEEE Control Systems Magazine**, Piscataway, NJ, USA, v.21, n.1, p.22–99, 2001.
- CASEY, B. J. Implementing Ethernet in the industrial environment. In: IEEE INDUSTRIAL APPLICATION SOCIETY ANNUAL MEETING, 1990. **Proceedings...** [S.l.]: IEEE Industry Applications Society, 1990. v.2, p.1469–1477.

CHOI, B.-Y.; SONG, S.; BIRCH, N.; HUANG, J. Probabilistic approach to switched Ethernet for real-time control applications. In: INTERNATIONAL CONFERENCE ON REAL-TIME SYSTEMS AND APLICATIONS, 7., 2000, Cheju Island, Korea. **Proceedings...** [S.l.]: IEEE Communications Society, 2000.

CLOUTIER, P.; MANTEGAZZA, P.; PAPACHARALAMBOUS, S.; SOANES, I.; HUGHES, S.; YAGHMOUR, K. DIAPM-RTAI Position Paper. In: REAL-TIME OPERATING SYSTEMS WORKSHOP, 2000, Milano, Italy. **Proceedings...** [S.l.]: Real-Time Linux Foundation, 2000. p.1-28.

CRAIG, J. J. **Introduction to Robotics Mechanics & Control**. Reading, MA, USA: Addison-Wesley, 1986.

FU, K. S.; GONZALEZ, R. C.; LEE, C. S. G. **Robotics Control, Sensing, Vision and Intelligence**. Singapore: McGraw-Hill, 1987.

FUNDA, J.; TAYLOR, R. H.; ELDRIDGE, B.; GOMORY, S.; GRUBEN, K. G. Constrained Cartesian motion control for teleoperated surgical robots. **IEEE Transactions on Robotics and Automation**, Piscataway, NJ, USA, v.12, p.453-465, 1996.

HARA, I.; HIRUKAWA, H. Distributed Robotics over the Internet. **IEEE Robotics and Automation Magazine**, Piscataway, NJ, USA, p.22-45, 2001.

HIRZINGER, G.; BRUNNER, B.; DIETRICH, J.; HEINDL, J. Sensor-based space robotics ROTEX and its telerobotic features. **IEEE Transactions on Robotics and Automation**, Piscataway, NJ, USA, v.9, n.5, p.649-663, 1993.

KALMAN, R. E. A New approach to Linear Filtering and Prediction Problems. **Transactions of the ASME - Journal of Basic Engineering**, [S.l.], p.35-45, 1960.

KWEON, S.; SHIN, K. G. Statistical Real-time Communication over Ethernet. **IEEE Transactions on Parallel and Distributed Systems**, Piscataway, NJ, USA, v.14, n.3, p.322-335, 2003.

LAGES, W. F.; ALT, G. H.; HENRIQUES, R. V. B. Controle em Tempo Real de Robôs Através de Rede Baseada no Protocolo IP. In: CONGRESSO TEMÁTICO DE APLICAÇÕES DE DINÂMICA E CONTROLE - DINCON 2003, 2., 2003, São José dos Campos, SP, Brasil. **Anais...** [S.l.]: SBMAC, 2003. (Série Arquimedes, v.2).

LIAN, F.; MOYNE, J. R.; TILBURY, D. M. Performance Evaluation of Control Networks: Ethernet, ControlNet and DeviceNet. **IEEE Control Systems Magazine**, Piscataway, NJ, USA, v.21, n.1, p.66-83, 2001.

NETFILTER. **The netfilter/iptables project**. Disponível em : <<http://www.netfilter.org>>. Acesso em: abr. 2003.

OGATA, K. **Engenharia de Controle Moderno**. Rio de Janeiro. RJ, Brasil: Prentince-Hall do Brasil, 1985.

- PAUL, R. P. **Robot Manipulators: mathematics, programming, and control.** Cambridge, MA, USA: MIT Press, 1981.
- PRESS, W. H.; TEUKOLSKY, S. A.; VETTERLING, W. T.; FLANNERY, B. P. **Numerical Recipes in C: the art of scientific computing.** 2nd.ed. Cambridge, MA, USA: Cambridge University Press, 1992.
- RAMAKRISHNAN, K.; YANG, H. The Ethernet capture effect: Analysis and solution. In: CONFERENCE ON LOCAL COMPUTER NETWORKS, 19., 1994, Minnesota, USA. **Proceedings...** [S.l.]: IEEE Computer Society, 1994. p.228–240.
- RIEDI, R.; COURSE, M.; RIBEIRO, V.; BARANIUK, R. A multifractal wavelet model with application to TCP network traffic. **IEEE Transactions on Information Theory**, Piscataway, NJ, USA, p.992–1018, 1999.
- SKEIE, T.; JOHANNESSEN, S.; BRUNNER, C. Ethernet in Substation Automation. **IEEE Control Systems Magazine**, Piscataway, NJ, USA, p.43–51, 2002.
- SPONG, M. W.; VIDYASAGAR, M. **Robot Dynamics and Control.** New York, USA: John Wiley & Sons, 1989.
- TANENBAUM, A. S. **Rede de Computadores.** Rio de Janeiro, Brasil: Campus, 1997.
- TOVAR, E.; VASQUES, F. Real-time Fieldbus communications using Profibus networks. **IEEE Transactions on Industrial Electronics**, Piscataway, NJ, USA, v.46, p.1241–1251, 1999.
- WALSH, G. C.; YE, H. Scheduling of Networked Control System. **IEEE Control Systems Magazine**, Piscataway, NJ, USA, v.21, n.1, p.57–65, 2001.
- XIAO, X.; NI, L. M. Internet QoS: A big picture. **IEEE Network**, Piscataway, NJ, USA, p.8–18, 1999.
- YE, H.; WALSH, G.; BUSHNELL, L. Wireless local area networks in the manufacturing industry. In: AMERICAN CONTROL CONFERENCE, 2000, Chicago, USA. **Proceedings...** [S.l.]: IEEE Control Systems Society, 2000. p.2363–2367.
- ZHANG, W.; BRANICKY, M. S.; PHILLIPS, S. M. Stability of Networked Control Systems. **IEEE Control Systems Magazine**, Piscataway, NJ, USA, v.21, n.1, p.84–99, 2001.